Alma Mater Studiorum · University of Bologna

School of Science
Department of Physics and Astronomy
Master Degree in Physics

Optical Diffraction Tomography: A Resolution Study

Supervisor:
Prof. Nico Lanconelli

Co-supervisor:
Prof. Jeroen Kalkman

Submitted by:
Valentino Recaldini

Academic Year 2019/2020

Abstract

In the past years, optical diffraction tomography (ODT) has been used both in cell imaging and to investigate the three-dimensional refractive index (RI) of large-scale (millimetre-sized) samples. In this technique, the projections at different illumination angles are acquired through digital holography (DH) and used to estimate the complex wave fields, which can be refocused with the aid of numerical diffraction algorithms. However, real extended specimens may not completely lie on a single plane. In this case, the (refocused) projections retain a certain amount of defocus which will affect the tomographic reconstruction.

For this reason, this thesis aims to study the spatial resolution of an ODT system when a point-like object is allowed to go in and out of focus and is reconstructed without numerical refocusing. Two-dimensional rotation and computational Fourier optics will be used to track and model defocus and lenses during the simulation of the projections. Spatial resolution will be assessed both qualitatively and quantitatively by numerically computing the full width at half maximum (FWHM) in relation to the maximum defocus to which a simulated point was subjected during acquisition.

Lastly, deconvolution is used to remove unwanted blur.

Contents

Chapter 1

Introduction

In the past years, optical diffraction tomography (ODT) has been employed in the investigation of the refractive index of millimetre-sized samples. Large specimens, however, seldom lie on a single image plane. Therefore, it is of interest to study the resolution of an ODT imaging system to attempt to improve it or to give a measure of the accuracy of any spatial measurement. Restoration of blurred images, if successful, improves spatial resolution and thus leads to more precise quantitative measurements.

Researchers at the Delft University of Technology (TU Delft) published a few papers on ODT in which they used the Mach-Zehnder interferometer during acquisition. The scope of this thesis is the investigation of the spatial resolution of an ODT system based on the Mach-Zehnder interferometer and the restoration of the reconstructed images.

Due to the COVID-19 crisis started in March 2020, we were unable to perform any experimental work in the laboratory, but all was not lost for this project. In fact, thanks to the theory of Fourier optics, it is possible to perform computer simulations of the key components in the interferometer and to analyse the results. We would like to remark that this thesis will focus on the effect of diffraction, interference and ideal lenses on the system and will assume an ideal, perfect detector that collects all the incoming light without any loss or noise. We will assume that all the other components—i.e. mirrors, beam splitter and beam expander—will not introduce additional, unwanted effects that might change the results.

This thesis is composed of two main parts. The first provides the theoretical background behind the computational work. The Fourier transform and its properties play a major role in this part, so they will be introduced first. Then, the model for image formation and the theory behind image restoration will be described. Computed tomography, which is the technique used to reconstruct three-dimensional objects from their two-dimensional projections, will be explained in detail. Light waves will be described next with particular focus on diffraction, interference and lenses. Then, two chapters will deal with holography and ODT to outline the key components and limitations behind these techniques.

The second part of this thesis focuses on the methods used for the practical application of the theory and the results obtained from the simulation work and deconvolution. In this part, the results of the optimisation of the processing pipeline are also presented.

Part I

Theoretical Background

Chapter 2

Fourier Transform

The Fourier transform is a powerful tool used to decompose a signal into its frequencies for either processing or analysis.

Let f be a real-valued or complex-valued integrable function such that $f : \mathbb{R}^N \to \mathbb{R}$ or $f : \mathbb{R}^N \to \mathbb{C}$, where N is a positive integer. Let $\mathbf{x}$ and $\boldsymbol{\xi}$ be N-dimensional vectors. Then, the N-dimensional Fourier transform of f is

$$\mathcal{F}f\left(\boldsymbol{\xi}\right) = \int_{\mathbb{R}^N} f\left(\mathbf{x}\right) e^{-i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} d\mathbf{x} \tag{2.1}$$

where $\mathbf{x} \cdot \boldsymbol{\xi}$ here is the Euclidean dot product and is sometimes written as $\langle \mathbf{x}, \boldsymbol{\xi} \rangle$. The dot product is also called scalar product and inner product. In this thesis, these terms will be used interchangeably.

For the sake of completeness, the expression for Euclidean dot product in $\mathbb{R}^N$ is reported below

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=0}^{N-1} x_i y_i \tag{2.2}$$

for any vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^N$. It is worth mentioning that it is positive definite and the right side of Equation (2.2) is 0 if $\mathbf{x}$ and $\mathbf{y}$ are perpendicular [1]. The norm, or length, of a vector $\mathbf{x}$ is

$$\|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}} \tag{2.3}$$

Let F be a function such that $F : \mathbb{R}^N \to \mathbb{C}$ and $F\left(\boldsymbol{\xi}\right) = \mathcal{F}f\left(\boldsymbol{\xi}\right)$. The inverse N-dimensional Fourier transform of F is defined as

$$\mathcal{F}^{-1}F\left(\mathbf{x}\right) = \int_{\mathbb{R}^N} F\left(\boldsymbol{\xi}\right) e^{i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi} \tag{2.4}$$

Please note that the inverse Fourier transform of F can be written as a Fourier transform,

$$\begin{aligned}\mathcal{F}^{-1}F(\mathbf{x}) &= \int_{\mathbb{R}^N} F\left(\boldsymbol{\xi}\right) e^{i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi} = \int_{\mathbb{R}^N} \overline{\overline{F}\left(\boldsymbol{\xi}\right)\ \overline{e^{i2\pi\mathbf{x}\cdot\boldsymbol{\xi}}}} d\boldsymbol{\xi} \\ &= \overline{\int_{\mathbb{R}^N} \overline{F}\left(\boldsymbol{\xi}\right) e^{-i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi}} = \overline{\mathcal{F}\left\{\overline{F}\right\}}(\mathbf{x}) = \overline{\mathcal{F}\overline{\mathcal{F}f}}(\mathbf{x})\end{aligned} \tag{2.5}$$

where the horizontal bar is used for the conjugation and the equivalence between the the two lines holds because $\boldsymbol{\xi} \in \mathbb{R}^N$ and because for any $z, w \in \mathbb{C}$

$$\overline{\overline{z}} = z \tag{2.6a}$$

$$\overline{zw} = \overline{z}\,\overline{w} \tag{2.6b}$$

$$\overline{z \pm w} = \overline{z} \pm \overline{w} \tag{2.6c}$$

Furthermore, with similar reasoning, we see that

$$\begin{aligned}\overline{\mathcal{F}\mathcal{F}f}(\mathbf{x}) &= \overline{\int_{\mathbb{R}^N} e^{-i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} \left(\int_{\mathbb{R}^N} f\left(\mathbf{x}\right) e^{-i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} d\mathbf{x}\right) d\boldsymbol{\xi}} \\ &= \int_{\mathbb{R}^N} e^{i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} \overline{\left(\int_{\mathbb{R}^N} f\left(\mathbf{x}\right) e^{-i2\pi\mathbf{x}\cdot\boldsymbol{\xi}} d\mathbf{x}\right)} d\boldsymbol{\xi} \\ &= \mathcal{F}^{-1}\{\overline{\mathcal{F}f}\}(\mathbf{x})\end{aligned} \tag{2.7}$$

Since this thesis mostly deals with images, image processing and computed tomography, we will present the expressions for the one- and two-dimensional Fourier transforms, their discrete formulae, and a few important properties.

2.1 One-Dimensional Fourier Transform

Let $f : \mathbb{R} \to \mathbb{C}$ and $F : \mathbb{R} \to \mathbb{C}$ be integrable functions. The one-dimensional Fourier transform of f is

$$\mathcal{F}f\left(u\right) = \int_{-\infty}^{+\infty} f\left(x\right) e^{-i2\pi xu} dx \tag{2.8}$$

and inverse is

$$\mathcal{F}^{-1}F\left(x\right) = \int_{-\infty}^{+\infty} F\left(u\right) e^{i2\pi xu} du \tag{2.9}$$

2.1.1 Properties

The Fourier transform has a few basic properties that are commonly used in signal processing and are therefore reported here for the one-dimensional case for the sake of completeness. Please keep in mind that they are valid for the N-dimensional transform as well.

Let $f(x)$, $F(u)$ and $h(x)$ be complex-valued integrable functions.

Some of the most important properties [2] of the Fourier transform are:

- **linearity**—the Fourier transform is linear,

  $$\mathcal{F}\left\{af\left(x\right)+bh\left(x\right)\right\}=a\mathcal{F}f\left(u\right)+b\mathcal{F}h\left(u\right) \tag{2.10}$$

  where a and b are scalars in $\mathbb{C}$.

- **shift**—any translation or time-shift by a constant $a\in\mathbb{R}$ introduces a phase shift in the frequency domain,

  $$\mathcal{F}\left\{f\left(x-a\right)\right\}=\mathcal{F}f\left(u\right)e^{-i2\pi ua} \tag{2.11}$$

  and multiplying f by $e^{i2\pi ax}$ shifts the frequencies by $a\in\mathbb{R}$,

  $$\mathcal{F}\left\{f\left(x\right)e^{i2\pi ax}\right\}=\mathcal{F}f\left(u-a\right) \tag{2.12}$$

- **scaling**—any scaling in the real domain stretches (or shrinks) the frequency spectrum accordingly and modifies the amplitude,

  $$\mathcal{F}\left\{f\left(ax\right)\right\}=\frac{1}{|a|}\mathcal{F}f\left(\frac{u}{a}\right) \tag{2.13}$$

  If $a=-1$, we are simply reversing the direction, and $\mathcal{F}\{f(-x)\}=\mathcal{F}f(-u)$.

- **Conjugation**—the Fourier transform of the conjugate of $f(x)$ is the conjugate of the Fourier transform of $f(-x)$, or

  $$\mathcal{F}\overline{f}(u)=\overline{\mathcal{F}f}(-u) \tag{2.14}$$

  Equation (2.14) can be combined with (2.13) into

  $$\mathcal{F}\{\overline{f}(-x)\}=\overline{\mathcal{F}f}(u) \tag{2.15}$$

  Furthermore, from Equations (2.7) and (2.15), we get

  $$\mathcal{F}\mathcal{F}f(x)=f(-x) \tag{2.16}$$

- **Fourier inversion theorem**—if, in addition to the previous hypotheses, f is continuous at x, and $F(u) = \mathcal{F}f(u)$, then

$$f(x) = \mathcal{F}^{-1}F(x) = \mathcal{F}^{-1}\mathcal{F}f(u) \tag{2.17}$$

- **convolution**—the convolution of two functions f and h is the inverse Fourier transform of the product of the Fourier transforms of f and h, or

$$(f * h)(x) = \int_{\mathbb{R}} f(t)\,h(x-t)\,dt = \mathcal{F}^{-1}\left\{\mathcal{F}f(u)\,\mathcal{F}h(u)\right\} \tag{2.18}$$

  However, the convolution of discrete signals via discrete Fourier transform is not as straightforward as Equation (2.18).

2.1.2 One-Dimensional Discrete Fourier Transform

Let $\{f_n\}$ be a sequence of N samples acquired at equally-spaced points x_n, where x_n could be a set of points on a line or a time sequence. The One-Dimensional (1D) Discrete Fourier Transform (DFT) of a discrete signal $\{f_n\}$ is [3]

$$F_m = \sum_{n=0}^{N-1} f_n e^{-i2\pi\frac{nm}{N}} \ , \ m = 0, 1, ..., N-1 \tag{2.19}$$

and its inverse (1D-IDFT) can be written as [3]

$$f_n = \frac{1}{N}\sum_{m=0}^{N-1} F_m e^{i2\pi\frac{nm}{N}} \ , \ n = 0, 1, ..., N-1 \tag{2.20}$$

Computer implementations of the 1D-DFT and 1D-IDFT, however, are not done through a straightforward implementation of the naive algorithms in Equations (2.19) and (2.20) as it does not scale well with the number of samples. Indeed, the computation of fhe DFT of a signal of length N has a complexity of $O(N^2)$.

To compute the DFT, we generally use the Fast Fourier Transform (FFT), which exploits properties of the DFT to reduce the number of complex multiplications during its computation [4]. Common algorithms are the radix-n (n integer), mixed-radix and prime-factor FFT. An additional speed-up can be achieved as well via parallelisation and vectorisation [4].

It is worth mentioning that all the radix-n implementations have their own requirements, which is $N = n^p$ with $n, p \in \mathbb{N}$. This, however, is not a limitation as the signal can be zero-padded (or zero-extended) to the nearest power of p greater than N and then sent to the FFT routine. This decreases the performance but provides better results than the DFT thanks to the huge speed-up the FFT provides: indeed, the radix-2 algorithm has a computational complexity of $O(N \log N)$. Please note that zero-padding before applying the Fourier transform interpolates the Fourier spectrum.

The implementation of the IDFT is usually done via FFT since the conjugate of the DFT of the conjugate of the sequence $\{F_m\}$ divided by N is the IDFT of F_m, or

$$\frac{1}{N}\overline{\sum_{n=0}^{N-1}\overline{F_n}e^{-i2\pi\frac{nm}{N}}} = \frac{1}{N}\sum_{n=0}^{N-1}F_n e^{i2\pi\frac{nm}{N}} = \mathrm{IDFT}\{F_m\}$$

which is the discrete version of Equation (2.5). The first equivalence follows from Equation (2.6), and the second is simply the definition of IDFT in (2.20).

Convolution

There are two types of convolution: linear and cyclic (or circular).

Let f and h be two sequences of length N and L. To convolve f and h via linear convolution using the FFT, both signals must be padded to at least $N + L - 1$, then Equation (2.18) can be used to compute the convolved signal.

On the other hand, to convolve two signals, f and h, of length N via cyclic convolution, Equation (2.18) can be directly applied.

Please note that the considerations made here for the discrete convolution can be extended to any dimension.

2.2 Two-Dimensional Fourier Transform

Let $f : \mathbb{R}^2 \to \mathbb{C}$ be an integrable function. The two-dimensional Fourier transform and of f and its inverse are defined as

$$F\left(u,v\right) = \mathcal{F}f\left(u,v\right) = \int\int_{\mathbb{R}^2} f\left(x,y\right)e^{-i2\pi(ux+vy)}dxdy \tag{2.21a}$$

$$f\left(x,y\right) = \mathcal{F}^{-1}F\left(x,y\right) = \int\int_{\mathbb{R}^2} F\left(u,v\right)e^{i2\pi(xu+yv)}dudv \tag{2.21b}$$

from Equations (2.1) and (2.4) where $\mathbf{x} = (x, y) \in \mathbb{R}^2$ and $\boldsymbol{\xi} = (u, v) \in \mathbb{R}^2$.

The 2D Fourier transform can be simplified if f is separable. We will say that f is separable in Cartesian coordinates if

$$f\left(x,y\right) = f_X\left(x\right)f_Y\left(y\right) \tag{2.22}$$

where f_X and f_Y are integrable functions that depend only on one coordinate (x and y, respectively). In that case, the x and y coordinates can be separated when we integrate and we have

$$\mathcal{F}f\left(u,v\right)=\mathcal{F}f_X\left(u\right)\mathcal{F}f_Y\left(v\right) \tag{2.23}$$

On the other hand, f will be separable in the polar coordinates (r,θ) if

$$f\left(r,\theta\right)=f_r\left(r\right)f_\theta\left(\theta\right) \tag{2.24}$$

where f_r depends only on the radial coordinate r and f_θ on the angular coordinate θ. In this case, two-dimensional operations can be still obtained via a series of one-dimensional operations, although it is not trivial [2]. Furthermore, if f does not depend on θ, we will say that f is circularly symmetric and write

$$f\left(r,\theta\right)=f_r\left(r\right) \tag{2.25}$$

For this particular class of functions, the Fourier transform and its inverse are equivalent [2], and we have

$$\mathcal{F}^{-1}\mathcal{F}f(r)=\mathcal{F}\mathcal{F}f(r)=f(r) \tag{2.26}$$

2.2.1 Two-Dimensional DFT

Let us sample f into NxM samples $f(n,m)$, and let $F(u,v)$ be its 2D discrete Fourier transform. Then, the 2D-DFT and its inverse are defined as

$$F\left(u,v\right)=\text{2D-DFT}\{f\left(n,m\right)\}=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}f\left(n,m\right)e^{-i2\pi\left(\frac{un}{N}+\frac{vm}{M}\right)} \tag{2.27a}$$

$$f\left(n,m\right)=\text{2D-IDFT}\{F\left(u,v\right)\}=\frac{1}{NM}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1}F\left(u,v\right)e^{i2\pi\left(\frac{un}{N}+\frac{vm}{M}\right)} \tag{2.27b}$$

The 2D-DFT and its inverse can be computed via FFT.

2.3 Power Spectrum

The Power Spectrum is often used in signal processing to analyse a signal based on its frequency content. Let $N\in\mathbb{N}$. The power spectrum of an integrable function $f:\mathbb{R}^N\to\mathbb{C}$ is

$$P_S=|\mathcal{F}f|^2 \tag{2.28}$$

However, when the visualisation of the power spectrum is required, it is common to display the common—base-10—logarithm of this expression because it can span several orders of magnitude and the DC term is often much larger than the rest of the spectrum.

In practice, the power spectrum of f gives the distribution over the frequency space and can be used to analyse the input, as shown in Fig. 2.1.

In particular, in Fig. 2.1, since the DC term can be computed by summing all the samples and the function is a sinusoidal oscillating about 0, we expect that the value of the power spectrum at 0 Hz will be 0. Furthermore, the function oscillates at a know frequency (1 Hz), and we see two peaks in its power spectrum, one for 1 Hz and one for -1 Hz.

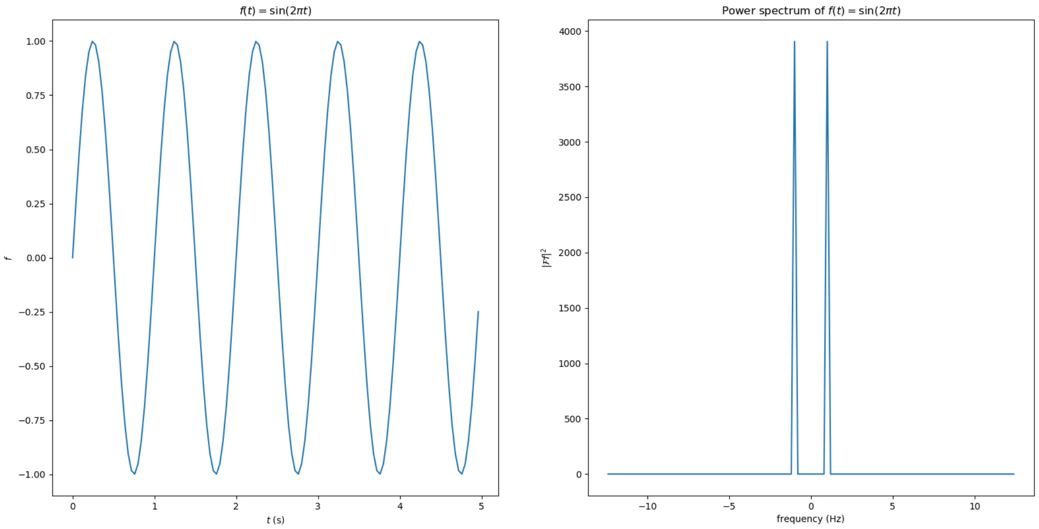


Fig. 2.1: Example of the power spectrum usefulness with a simple sinusoidal function oscillating at a known frequency.

Equation (2.28) can be rewritten to offer more insight on the meaning of the power spectrum:

$$|\mathcal{F}f|^2 = \mathcal{F}f\overline{\mathcal{F}f} = \mathcal{F}\{f(\mathbf{x}) * \overline{f}(-\mathbf{x})\} \tag{2.29}$$

where $\mathbf{x} \in \mathbb{R}^N$ and the last equivalence follows from Equation (2.15). The convolution in the Fourier transform in the last equivalence of Equation (2.29) is the cross-correlation of f with itself, also known as autocorrelation. Therefore, the autocorrelation of f is the inverse Fourier transform of the power spectrum of f, or

$$f(\mathbf{x}) * \overline{f}(-\mathbf{x}) = \mathcal{F}^{-1}\left\{|\mathcal{F}f|^2\right\} \tag{2.30}$$

Please note that if $f : \mathbb{R}^2 \to \mathbb{R}$ and Equation (2.25) holds (i.e. f is circularly symmetric), then $f(x, y) = \overline{f}(-x, -y)$ and $f * f = \mathcal{F}^{-1}\{|\mathcal{F}f|^2\}$, which also implies that the Fourier transform of f is real-valued. Indeed, we have

$$f : \mathbb{R}^2 \to \mathbb{R},\ f_r : \mathbb{R} \to \mathbb{R} \ \mid \ f(r, \theta) = f_r(r) \Rightarrow \overline{\mathcal{F}f} = \mathcal{F}f \tag{2.31}$$

where (r, θ) is the set of polar coordinates. The relation in (2.31) follows from Equations (2.15) and (2.29).

Chapter 3

Image

For the scope of this thesis, we will define an image as the interaction of a source with an object and the apparatus used for recording, and we will limit our discussion to transmission imaging. We will call the source-apparatus system the *imaging system* and say that it transforms an input I into an output O via an operator $\mathcal{S}$, or

$$O\left(x,y\right)=\mathcal{S}\{I\left(\xi,\eta\right)\} \tag{3.1}$$

where (x,y) and (ξ,η) are the coordinates on the detector and on the object respectively. If $\mathcal{S}$ is known, Equation (3.1) can be used to predict the effect of the system on any ideal input.

An imaging system is said to be *linear* if, for any constants $\alpha,\beta\in\mathbb{C}$ and inputs I_1 and I_2, we have [2]

$$\mathcal{S}\{\alpha I_1\left(\xi,\eta\right)+\beta I_2\left(\xi,\eta\right)\}=\alpha\mathcal{S}\{I_1\left(\xi,\eta\right)\}+\beta\mathcal{S}\{I_2\left(\xi,\eta\right)\} \tag{3.2}$$

In that case, Equation (3.1) can be rewritten via a superposition integral [2, 3], namely

$$O\left(x,y\right)=\int\int_{\mathbb{R}^2}I\left(\xi,\eta\right)h\left(x,y;\xi,\eta\right)d\xi d\eta \tag{3.3}$$

where $h\left(x,y;\eta,\xi\right)$ is the function that describes the effect of the system on the input and is sometimes called impulse response or, in optics, point spread function (PSF). If $h(x,y;\xi,\eta)$ depends only on the distances $(x-\xi)$ and $(y-\eta)$ between the points, we will say that the PSF is space-invariant—or, sometimes, shift-invariant—and write

$$\begin{aligned}O\left(x,y\right)&=\int\int_{\mathbb{R}^2}I\left(\xi,\eta\right)h\left(x-\xi,y-\eta\right)d\xi d\eta\\&=\left(I*h\right)\left(x,y\right)\end{aligned} \tag{3.4}$$

which is the convolution of the impulse response with the input. Equation (3.4) implies that the input $I(\eta, \xi)$ can theoretically be recovered, providing that h is known. In that case, we will say that we deconvolve O with h to find I.

The effect of noise into the output image has been neglected until now. However, real images are often affected by random fluctuations. To consider this, we use an additive model and write

$$\tilde{O}(x, y) = O(x, y) + n(x, y) \tag{3.5}$$

where $n(x, y)$ represents the noise of the image and $O(x, y)$ can be computed with Equation (3.1) or any valid simplified formula depending on the assumptions on the imaging system.

3.1 Digital Images

Nowadays, image processing is mostly done with a computer because it offers a much bigger variety of operations. Furthermore, digital images are not subject to corruption due to physical damage and can be copied and stored without difficulty.

Any physical image can be described as a matrix of N rows and M columns. Let us define the row index $i = 0, 1, ..., N - 1$ and the column index $j = 0, 1, ..., M - 1$. The indices i and j represent the y and x coordinates at which the input is sampled.

Furthermore, we assume two constant sampling intervals: one between any two adjacent rows i and $i + 1$ (pixel height, Δy), and the other between two adjacent columns j and $j + 1$ (pixel width, Δx). It is common, but not required, that the pixel height and width are identical. These two numbers limit the spatial resolution of the system but do not entirely describe it.

3.2 Linear Imaging System: a Vector-Matrix Formulation

Let $\mathcal{S}$ be a linear operator for which Equation (3.3) is satisfied for any input-output pair $f, g : \mathbb{R}^2 \to \mathbb{C}$. The discrete signals of f and g can be represented by two $N \times M$ matrices, which can be translated to two vectors $\mathbf{f}, \mathbf{g} \in \mathbb{C}^Q$ with $Q = NM$ and $N, M \in \mathbb{N}$.

Since $\mathcal{S}$ is a linear operator, we can write Equation (3.3) for a discrete signal as a matrix-vector product,

$$\mathbf{g} = \mathbf{H}\mathbf{f} \tag{3.6}$$

where $\mathbf{H}$ is a complex-valued matrix with Q rows and Q columns, and it represents the linear transformation caused by the system $\mathcal{S}$. Therefore, $\mathbf{H}$ has Q^2 elements.

Real systems, however, are also characterised by the presence of noise. Following Equation (3.5) and using $\mathbf{n}$ as the vector representation of the matrix that describes the noise, we have

$$\mathbf{g} = \mathbf{Hf} + \mathbf{n} \tag{3.7}$$

Straightforward implementation of the linear equations above is not recommended, however, and it is instead discouraged due to the large computational and memory requirements. Indeed, for an image with $512 \times 512 = 2^{18}$ pixels, $\mathbf{H}$ is defined as a matrix with 2^{36} elements. Please note that nowadays it is not uncommon to find detectors and cameras with larger dimensions that even reach tens of millions of pixels.

Space-Invariant Blurring

To express a space-invariant linear system as the one in Equation (3.4) for a discrete signal as in Equation (3.6), we have to write a matrix $\mathbf{H}$ that defines the convolution with a two-dimensional kernel h. First, however, we have to define a set of matrices called Toeplitz matrices and a special subset, the circulant matrices.

Let $\mathbf{T}$ be a $Q \times Q$ matrix, where Q is a positive integer. $\mathbf{T}$ is called a Toeplitz matrix if

$$\mathbf{T} = \begin{bmatrix} t_0 & t_{-1} & \dots & t_{-(Q-1)} \\ t_1 & t_0 & \dots & t_{-(Q-2)} \\ \vdots & & \ddots & \\ t_{Q-1} & & & t_0 \end{bmatrix} \tag{3.8}$$

The elements in $\mathbf{T}$ can also be expressed as $t_{i,j} = t_{i-j}$ provided $t_{i-j} \in \{t_q \mid -(Q-1) \leq q \leq Q-1\}$.

A particular kind of Toeplitz matrices are the *circulant* matrices, for which we have

$$\mathbf{C} = \begin{bmatrix} c_0 & c_{Q-i} & \dots & c_1 \\ c_1 & c_0 & \dots & c_2 \\ \vdots & \ddots & \ddots & \vdots \\ c_{Q-i} & \dots & c_1 & c_0 \end{bmatrix} \tag{3.9}$$

Please note that the transpose of this matrix can be considered a circulant matrix as well.

Let $\mathbf{A}$ be a block matrix composed of matrix blocks $\mathbf{A}_{i,j}$. If the blocks are arranged following the scheme in Equation (3.8) where each element $t_{i,j}$ is now a block matrix, then $\mathbf{A}$ is called block-Toeplitz matrix. Furthermore, if each individual block $\mathbf{A}_{i,j}$ is also a Toeplitz matrix, then $\mathbf{A}$ is a doubly block-Toeplitz matrix. Analogously, a block matrix whose blocks follow the scheme for the circulant matrix (see Equation (3.9)) is called block-circulant, and if each block is also circulant we have a block-circulant matrix with circulant blocks, or a doubly-circulant matrix.

These kinds of matrices can be used to describe a linear space-invariant imaging system as a linear operator. Therefore, they can represent the convolution with one- and two-dimensional kernels. Now, we will describe how this matrices can be constructed for a two-dimensional kernel.

For a linear convolution with a two-dimensional PSF, we first need to extend the two-dimensional signal and kernel by using the same consideration previously made for the DFT or FFT. Then for each row of the kernel we write a Toeplitz block and arrange all the blocks into a doubly block-Toeplitz matrix.

To construct a KxK Toeplitz block $\mathbf{G}$ from a row $\mathbf{a}$ of length K extracted from the zero-padded kernel, we can use the scheme in Equation (3.8):

Step 1: the first column ($i = 0$) of $\mathbf{G}$ to $\mathbf{a}^\intercal$, which is the transpose of $\mathbf{a}$;

Step 2: for each column $i = 1, 2, ..., K - 1$, do a circular shift by i elements on $\mathbf{a}^\intercal$ and use the new column vector to fill the i-th column of $\mathbf{G}$.

For a cyclic (circular) convolution with a two-dimensional PSF, we use a doubly block-circulant matrix: i.e. a block matrix $\mathbf{C}$ whose blocks are arranged according to (3.9) and are circulant matrices as well. The construction of the matrix is similar, but zero-padding is not performed if both the kernel and the signal have the same length (in both dimensions).

Convolution with a one-dimensional PSF can be constructed in a similar fashion. The resulting matrix, however, will not be a block matrix.

A doubly block-circulant matrix is sometimes used instead of $\mathbf{H}$ for a space-invariant PSF as it offers important properties in optimisation problems and the matrix-vector product $\mathbf{Hf}$ is represented by a cyclic convolution with a two-dimensional PSF. Please note, however, that, for discrete signals, Equation (3.4) can be efficiently computed by exploiting (2.18) and the 2D-FFT routines, and matrix-vector notation is generally useful for further insight in the imaging model or to find a solution to deconvolution with linear algebra notation, which may then be translated into a series of Fourier transforms in this case.

3.3 Spatial Resolution

From Nyquist's theorem, if a signal $g(t)$ is sampled at discrete intervals, the signal is correctly represented by the sampled sequence if and only if the maximum frequency in $g(t)$ is not higher than half the sample rate f_s detectable by the detector, or

$$u_{g,max} \leq \frac{1}{2} f_s \tag{3.10}$$

where $u_{g,max}$ is the maximum frequency in $g(t)$ and f_s can be computed as

$$f_s = \frac{1}{\tau} \tag{3.11}$$

where τ is the sampling interval. Therefore, the Nyquist frequency is

$$f_N = \frac{1}{2\tau} \tag{3.12}$$

where τ can be either the pixel width or the pixel height, depending on the spatial direction.

If Equation (3.10) is not followed, the digital signal will be distorted by aliasing due to the inability to detect and represent the correct frequencies. Digital images are, of course, affected by this theorem as they are two-dimensional discrete signals. However, Nyquist's theorem is not sufficient to describe spatial resolution and represents only the upper limit of the detector.

Generally, the resolution of the system is computed from its PSF. Indeed, this particular function is used to compute two important metrics—the full width at half maximum (FWHM), which is the width of the PSF at half peak value, and the modulation transfer function (MTF) which is calculated as the modulus of the Fourier transfer of the PSF, or

$$\mathrm{MTF}\left(u, v\right) = \left|\mathcal{F}h\left(u, v\right)\right| \tag{3.13}$$

where u and v are the discrete spatial frequencies and are usually computed as

$$u = \frac{m}{M\Delta x} \quad , \quad v = \frac{n}{N\Delta y} \tag{3.14}$$

where M and N are, respectively, the detector width and height in pixels, and $m, n \in \mathbb{Z} : |m| \leq M/2,\ |n| \leq N/2$.

The MTF is usually normalised, i.e. the maximum value (DC term) is 1, and thus we will work with normalised MTFs. $\mathcal{F}h$ is sometimes called transfer function of the system. We will usually denote such a function with H.

The MTF of the system is the product of all the MTFs of the elements of the apparatus. In the spatial domain, this means that the PSF of the system is the convolution of the PSFs of all of its parts.

This has some interesting consequences. Unless the normalised MTF of a component is 1 for all the frequencies, that component will degrade the spatial resolution of the whole system, which means that, in general, the resolution worsens as the number of parts increases. If one element in the apparatus has an MTF that quickly goes to 0, the MTF of the whole system will quickly reach 0, cutting all the high frequencies.

Furthermore, the broadness of the MTF affects the sharpness of the edges. Indeed, a broader MTF will have sharper edges because these generate higher frequencies in the Fourier transform. For this reason, sharpness is also affected by how quickly the MTF reaches 0. Indeed, the MTF practically weights the contribution of each set of frequencies (u, v) of the input image.

3.4 Image Restoration

Restoration is the process of improving any degraded image $\tilde{O}$ by filtering it. It may be used to remove outliers and noise or recover an estimate $\hat{I}$ of the real input I by taking the imaging system into account. In this section, we will focus on image deconvolution.

Please note that, throughout this section, the coordinates will be dropped in the notation and that for any $f, g : \mathbb{R}^2 \to \mathbb{C}$ integrable, we will $\mathcal{F}f(u, v) = \mathcal{F}f$ and $\mathcal{F}g(u, v) = \mathcal{F}g$. Furthermore, any operation between the Fourier transforms of f and g will be considered frequency-wise unless otherwise stated. For example, $\mathcal{F}f\mathcal{F}g = \mathcal{F}f(u, v)\mathcal{F}g(u, v)$.

3.4.1 Inverse Filtering

Inverse filtering is the easiest deconvolution routine we can employ to recover the image, and is modelled as

$$\hat{I} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\tilde{O}}{H}\right\} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\tilde{O}}{\mathcal{F}h}\right\} \tag{3.15}$$

where the deconvolution is performed by exploiting Equation (2.18). However, substituting $\tilde{O}$ with Equation (3.5) and using the linearity of the Fourier transform (2.10), we find that

$$\mathcal{F}\hat{I} = \frac{\mathcal{F}O}{H} + \frac{N}{H} = \mathcal{F}I + \frac{N}{H} \tag{3.16}$$

where $N = \mathcal{F}n$ is the Fourier transform of the noise in the image. This leads to the conclusion that inverse filtering of a real image will never recover the original image I because noise is not taken into account [3]. Preprocessing the image with some de-noising algorithm might improve $\hat{I}$, but direct inversion is usually avoided.

3.4.2 Richardson-Lucy Deconvolution

Richardson-Lucy's [5] algorithm is an iterative routine that can be used to deconvolve an image. For any iteration $k = 1, ..., N$, the estimate of I is

$$\hat{I}_k = \hat{I}_{k-1} \left(\left(\frac{\tilde{O}}{\hat{I}_{k-1} * h} \right) * h_m \right) \tag{3.17}$$

where h_m is the mirrored PSF on both axes—namely, $h_m(x, y) = h(-x, -y)$; we invert the spatial direction to obtain h_m. $\hat{I}_0$ is a uniform image.

For any convolution, we can use Equation (2.18), which relates the convolution to a series of Fourier transforms. Please note that the PSF h is usually real-valued and therefore $\overline{h}(-x, -y) = h(-x, -y)$. in this case, the convolution with h_m is simply a multiplication with the conjugate Fourier transform of H in the Fourier domain.

Moreover, please note that if h is circularly symmetric (i.e. it follows Equation (2.25)), $h_m = h$. If the PSF is not only circularly symmetric but also real-valued, then we also have (2.31).

In Equation (3.17), we see that the new estimate $\hat{I}_k$ is computed by multiplying the current estimate $\hat{I}_{k-1}$ with a correction factor, which depends on the degraded image and the estimated degraded image.

We can further analyse the correction factor. From $\frac{\tilde{O}}{\hat{I}_k * h}$, we can see that the output is compared against the blurred estimate. If the PSF h closely describes the effect of the system on the input and noise is negligible, then we expect that the comparison of the blurred image against the blurred estimate will be constant when $\hat{I}_k$ is a scaled version of the true input I, namely $I = c\hat{I}_k$ with $c \in \mathbb{C}$. In that case, the correction factor is a constant. This can easily be verified in the one-dimensional case, and the result can be extended to the n-dimensional case. Indeed, if $f : \mathbb{R} \to \mathbb{C}$ and $f(x) = c$, we have

$$(f * h)(t) = c \int_{-\infty}^{+\infty} h(x)dx = const$$

where we have exploited the commutativity of the convolution,

$$(f * h)(t) = \int_{-\infty}^{+\infty} f(x)h(t-x)dx = \int_{-\infty}^{+\infty} f(t-x)h(x)dx = (h * f)(t)$$

From this reasoning, it follows that, under these conditions, if we reach a scaled version of I at the iteration $\tilde{k}$, we expect that the estimate I_k will only be rescaled $\forall k > \tilde{k}$.

Unfortunately, Richardson-Lucy's deconvolution algorithm has two inherent drawbacks. The first is caused by the division in the correction factor. In fact, if $\tilde{O}(x, y)$ is zero for some $(x, y) \in \mathbb{R}^2$, then we have the undefined case $\frac{0}{0}$ as I_k approximates the true input I with increasing precision. This problem, however, may be circumvented by adding a small constant to $\tilde{O}$ and remembering that the convolution follows Equation (3.2) since it is linear and that the convolution with a constant is a constant. Therefore, this operation does not change the system, which can still be described by the same PSF.

The second drawback is similar to that for the inverse filter. While Richardson-Lucy's deconvolution attempts to reconstruct the input iteratively from a known PSF, it does not take the effect of noise into account.

3.4.3 Wiener Filtering

Wiener filtering, or Minimum Mean Square Error Filtering, is a type of inverse filtering that incorporates noise into the algorithm and attempts to minimise an error measure that is given by the expected value of $(I - \hat{I})^2$, where I and $\hat{I}$ are, as usual, the true image and its estimate.

We will use the same notation that was introduced in (3.4) and (3.5). Let $S_n = |\mathcal{F}n|^2$ and $S_I = |\mathcal{F}I|^2$, and let H be the Fourier transform of the PSF.

Then, the Fourier transform of the estimate $\hat{I}$ of the true image is [3]

$$\begin{aligned} \mathcal{F}\hat{I} &= \frac{\overline{H} S_I}{S_I |H|^2 + S_n} \mathcal{F}\tilde{O} = \\ &= \frac{1}{H} \frac{|H|^2}{|H|^2 + S_n/S_I} \mathcal{F}\tilde{O} \end{aligned} \tag{3.18}$$

which means that the deconvolution of a noisy signal can be written as a convolution with a pseudo-inverse kernel. In fact, if we define $k = \mathcal{F}^{-1}K$ such that $K = \frac{1}{H} \frac{|H|^2}{|H|^2 + S_n/S_I}$, then the Wiener filter in Equation (3.18) can be written as

$$\hat{I} = \tilde{O} * k \tag{3.19}$$

by applying the inverse Fourier to both sides of (3.18) and using (2.18) and the commutativity of the convolution.

This result has also been used by Xu et al. [6] to build a deep convolutional neural network for image deconvolution.

Equation (3.18) depends on the spectrum of the true image, which is generally not known nor is S_n. In those cases, the Wiener filter cannot be estimated directly, and we may approximate it by substituting S_n/s_I with a constant $q \in \mathbb{R}$,

$$\mathcal{F}\hat{I} = \frac{1}{H}\frac{|H|^2}{|H|^2 + q}\mathcal{F}\tilde{O} \tag{3.20}$$

This approximation, however, might not be a suitable option.

Another approach to improve Equation (3.20) is the Constrained Least Squares (CLS). Let f, g be an input-output pair and $\mathbf{f}$ and $\mathbf{g}$ their vector representation for which (3.5) and (3.7) are valid. Let M and N be the image width and height in pixels, respectively. To tackle this problem, we start by expressing the system with Equation (3.7) and by defining a criterion, or a measure of smoothness, depending, for example, on the Laplacian,

$$J = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|\nabla^2 f(x,y)|^2 \tag{3.21}$$

that must be minimised and is subject to the constraint

$$||\mathbf{g} - \mathbf{Hf}||^2 = ||\mathbf{n}||^2 \tag{3.22}$$

where $\mathbf{n}$ is the vector representation of the noise in the image.

The Laplacian operator can be written as a matrix,

$$\mathbf{L} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{3.23}$$

and the Fourier transform of $\hat{f}$, which the estimate of f, is given by [7]

$$\mathcal{F}\hat{f} = \frac{\overline{H}}{|H|^2 + \gamma|\mathcal{F}p|^2}\mathcal{F}g \tag{3.24}$$

where p is an $N \times M$ matrix that is 0 everywhere except in the central region in which $\mathbf{L}$ has been embedded [3]. Furthermore, γ is a parameter that must be adjusted to satisfy the constraint (3.22) and may be computed iteratively.

Please note that (3.20) and (3.24) both reduce to inverse filtering when the constant (namely, q or γ) is set to 0.

3.4.4 Blind Deconvolution

Sometimes the PSF is not known or cannot be estimated. For example, we might have only partial or no information about the imaging system, or the sample might have accidentally moved during its acquisition. Therefore, algorithms that assume complete knowledge of the impulse response of the system cannot be employed to retrieve the true signal.

However, all is not lost. Let us assume that the blur is space-invariant. In fact, there are some strategies that aim to find a pair of functions I and h that satisfies either Equation (3.4) or (3.5). Those are grouped under the label *blind deconvolution* or *blind image restoration*. When partial information of the PSF is available and incorporated into the restoration algorithm, we sometimes talk about semi-blind deconvolution.

The main requirement on I and h for a successful blind deconvolution is that both the input and the PSF must be irreducible. To justify this limitation, let us consider a set of functions $f_1, f_2, ..., f_N$ with $N \in \mathbb{N}$ and $N > 1$. Now, if I (or alternately h) can be expressed as $f_1 * f_2 * ... * f_N$, then the blind deconvolution algorithm cannot retrieve I and h unambiguously as any combination is a valid solution. This is straightforward from the associativity of the convolution,

$$\begin{aligned}(f_1 * ... * f_N) * h &= f_1 * (f_2 * ... * f_N * h) \\ &= (f_1 * ... * f_n) * (f_{n+1} * ... * f_N * h)\end{aligned}$$

for any $n \in \mathbb{N}$ such that $1 < n < N$.

However, even under the assumption of irreducibility of I and h, blind deconvolution is not without limitation. In fact, it is an ill-posed problem, as a small perturbation in the measured data may drastically change the result, and the solution might not be unique. Furthermore, even in this case noise prevents exact image restoration because—at best—only statistical information about noise is known.

The expected output of blind deconvolution is a shifted and scaled version of the true image,

$$\hat{I}(x, y) = qI(x - a, y - b) \ , \quad q, a, b \in \mathbb{R} \tag{3.25}$$

where the three constants may be retrieved by employing additional constraints [8]. For example, it is not uncommon to introduce further limitations on the image and the PSF such as non-negativity or to restrict the restoration algorithm to certain regions by introducing the supports of the image and the PSF.

While there are many approaches to this problem, as shown in Kundur's review [8], we will focus on two iterative algorithms to provide a better insight on the method. While both will use an alternating minimisation technique, one will attempt to minimise a global cost function, while the other comes from a modification of Richardson-Lucy's deconvolution.

Deconvolution via Simulated Annealing

Simulated Annealing (SA) [9] is an optimisation techniques that originates from statistical mechanics considerations and aims to minimise a cost function. Its strength relies in its ability to escape local minima by jumping to a state with higher cost depending on pseudo-random processes as the system is cooled.

Let k be a positive integer representing the iteration, $\mathbf{x}_{k-1}, \mathbf{x}_k \in \mathbb{R}^M$ (with $M \in \mathbb{N}$) be the state of the system at the previous and at the current iterations respectively. Let $J : \mathbb{R}^M \to \mathbb{R}$ be the chosen cost function, T_0 be the initial temperature and $f(T)$ a cooling schedule such that $f(T)$ is a monotonically decreasing function. The temperature must always be positive.

Let d be a positive small real number that will be used to stochastically update the state.

Now, we have to define two integers, N_c and N_s, which represent the number of cycles and the number of scans, respectively. The former represents the number of times the temperature of the system has to be updated, and the latter defines the number of scans at constant temperature.

Then, the algorithm can be summarised into few steps:

Step 1: the initial state $\mathbf{x}_0$ is loaded from an available guess or generated randomly. Set $n_c = 0$, $n_s = 0$ and $t_c = T_0$;

Step 2: increase n_c by 1 and update t_c according to $f(t_{c-1})$;

Step 3: increase n_s by 1;

Step 4: generate M pseudo-random number uniformly distributed in $[-d, d]$ and store them in $\mathbf{n} \in \mathbb{R}^M$. Define $\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{n}$;

Step 5: define the variation of cost $\Delta J \coloneqq J(\mathbf{x}_k) - J(\mathbf{x}_{k-1})$. If $\Delta J \leq 0$ (i.e. if the current cost is smaller than that of the previous state) accept the new state. Otherwise, generate a pseudo-random number r from a uniform distribution in $[0, 1]$. If $r \leq e^{-\frac{\Delta J}{t_c}}$ accept the new state; otherwise, reject it and set $\mathbf{x}_k = \mathbf{x}_{k-1}$.

Step 6: if $n_s < N_s$, jump to **Step 3**. Otherwise, go to the next step;

Step 7: if $n_c < N_c$, set $n_s = 0$ and jump to **Step 2**. Otherwise, stop.

As it can clearly be seen from **Step 5** of the algorithm, the probability of jumping to a state with higher cost depends on the temperature of the current cycle. Therefore, early cycles will have a higher chance to make this jump. Furthermore, this probability is intimately linked to the initial temperature T_0. In fact, we can see that the probability of the jump tends to 1 (certain event) if T_0 tends to infinity, which implies that the initial parameter must be calibrated.

SA is guaranteed to converge provided that the temperature does not decrease any faster than [10]

$$T_k = \frac{T_0}{\ln(k+1)} \ , \quad k = 1, 2, ... \tag{3.26}$$

which, unfortunately, might be quite slow.

To overcome this drawback, it is not uncommon to prefer a faster schedule like

$$T_k = \alpha T_{k-1} \ , \quad k = 1, 2, ... \tag{3.27}$$

which, however, may lead to a local minimum and thus fail to reach an optimal solution.

In 1990, McCallum [11] applied SA to blind image deconvolution with a few variations. He limited the variation of the state to the pixels within the supports for the estimate of the true image $\hat{f}$ and the PSF $\hat{h}$ in order to reduce the number of variables that has to be optimised.

Furthermore, McCallum adopted an alternating minimisation strategy by updating first $\hat{f}$ and then $\hat{h}$ at each scan. The constraints on $\hat{f}$ and $\hat{h}$ were incorporated in the algorithm just before the evaluation of the cost, thus making this method very flexible.

However, the choice of T_0 and d is not straightforward and wrong values may lead to local minima or require a larger number of iterations to converge to the optimal solution.

The computational time for the algorithm is very high, which makes it impractical for images of average size [8].

(Semi-)Blind Richardson-Lucy Deconvolution

McCallum's SA blind deconvolution provides an example of iterative algorithm that requires the definition of a cost function. However, there are other options that lift this requirement.

Fish *et al.* [12] introduced a blind image restoration algorithm based on Richardson-Lucy's deconvolution. Even in this case, the estimate of the PSF and the true image are updated alternately. However, differently from other cases, each update may take several iterations.

Let K be the number of iterations of the algorithm, and let J be the number of iterations for a single update.

Let g be the degraded image. Let $\hat{f}$ and $\hat{h}$ be the estimate of the true image and the PSF, and let $\hat{f}_0$ and $\hat{h}_0$ be the initial guesses. These could either be uniform images or images obtained from partial knowledge of the true image and PSF.

Since at each iteration the algorithm must produce an estimate in an iterative fashion, when this is the case, we will write the second index after a semicolon. Furthermore, we will consider $\hat{f}_{k;0} = \hat{f}_{k-1;J}$, which implies $\hat{f}_{1;0} = \hat{f}_0$. Now we can define the estimates at the k-th iteration with $k = 1, 2, ..., K$.

Then, for $j = 0, 2, ..., J - 1$, we first produce an estimate of the PSF at the k-th iteration,

$$\hat{h}_{k;j+1}(x,y) = \left(\left(\frac{g(x,y)}{\hat{h}_{k,j}(x,y) * \hat{f}_{k-1}(x,y)} \right) * \hat{f}_{k-1}(-x,-y) \right) \hat{h}_{k;j}(x,y) \quad (3.28)$$

Now, we drop the second index on $\hat{h}_{k,J}(x,y)$ and write $\hat{h}_k(x,y)$. For $j = 0, 2, ..., J - 1$, we produce an estimate for the true image at the k-th iteration,

$$\hat{f}_{k;j+1}(x,y) = \left(\left(\frac{g(x,y)}{\hat{f}_{k;j}(x,y) * \hat{h}_k(x,y)} \right) * \hat{h}_k(-x,-y) \right) \hat{f}_{k;j}(x,y) \quad (3.29)$$

Equations (3.28) and (3.29) lead to a few interesting considerations.

First, the structure of both equations is exactly the one from Richardson-Lucy's deconvolution, and therefore all the remarks done previously on Richardson-Lucy's algorithm are still valid. Secondly, the first equation is an attempt to deconvolve the PSF from an estimate of the true image, while the second equation, like Richardson-Lucy's deconvolution, aims to restore the true image with the estimate of the PSF.

3.4.5 Space-Variant Deconvolution

Sometimes, the blur is not constant over the image and varies with the pixel location instead, which renders Equation (3.4) invalid. In these cases, we say that the PSF is space-variant, and we describe the system with the superposition integral in Equation (3.3). For this reason, usual deconvolution methods cannot be applied directly. However, it is still possible to retrieve an estimate of the true image.

Nagy and O'Leary's Space-Variant Deconvolution

Nagy and O'Leary [13] proposed a solution to this problem. They assumed that an image that was degraded by space-variant blur could be split into p subregions and that the PSF of each region was roughly space-invariant and could be measured. In this case, the system can be represented and restored with a series of Fourier transform.

Let $\mathbf{H}$ be a matrix that represents the impulse response of the system, $\mathbf{f}$ the vector representation of the true image and $\mathbf{g}$ the degraded image, which is measured. Let the matrix representation of $\mathbf{g}$ and $\mathbf{f}$ be two $N \times M$ matrices with $N, M \in \mathbb{N}$. Then, exploiting the matrix-vector notation in Equation (3.6) for the linear systems like (3.3), Nagy and O'Leary's assumption on $\mathbf{H}$ can be easily rewritten as

$$\mathbf{H} = \sum_{k=1}^{p} \mathbf{D}_k \mathbf{H}_k \tag{3.30}$$

where each $\mathbf{H}_k$ is either a doubly block-Toeplitz or doubly block-circulant matrix and each $\mathbf{D}_k$ is a diagonal matrix with non-negative diagonal entries such that $\sum_{k=1}^{p} \mathbf{D}_k = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix.

Equation (3.6) now becomes

$$\mathbf{g} = \sum_{k=1}^{p} \mathbf{D}_k \mathbf{H}_k \mathbf{f} \tag{3.31}$$

which can be rewritten as a weighted sum of p two dimensional convolutions by exploiting the associativity of matrix multiplication.

From Equation (3.31), it can be see that the k-th convolution $\mathbf{H}_k\mathbf{f}$ is masked with a diagonal matrix $\mathbf{D}_k$. Indeed, each element on the diagonal of $\mathbf{D}_k$ represents a pixel in the image. If that element is 0, then the k-th convolution will not affect it. The constraint on the non-negativity of the coefficients in $\mathbf{D}_k$ and the sum to the identity can also be interpreted as a linear interpolation of the convolutions, which may improve the results on the outer borders of each region.

It can be verified that each $\mathbf{D}_k$ can be represented either as a vector like $\mathbf{f}$ by considering only the elements on the diagonal or as an image by rearranging the elements on the diagonal into an $N \times M$ matrix. For the sake of simplicity, let $\mathbf{d}_k$ be the vector generated by collecting the elements on the diagonal of $\mathbf{D}_k$, and let $\mathbf{g}_k = \mathbf{H}_k\mathbf{f}$. Then, the matrix-vector multiplication of $\mathbf{D}_k$ with $\mathbf{H}_k\mathbf{f}$ can be easily computed with an element-wise (Hadamard) product of $\mathbf{D_k}$ with $\mathbf{g}_k$, which is commonly denoted with $\mathbf{d}_k \odot \mathbf{g}_k$. Alternately, if $g_k(x, y)$ is the image whose vector representation is $\mathbf{g}_k$ and $d_k(x, y)$ is image that generates $\mathbf{d}_k$, then

$$\mathbf{d}_k \odot \mathbf{g}_k = d_k(x,y)g_k(x,y)$$

Now, using $f(x,y)$ and $h_k(x,y)$ for the image and the PSF respectively, we can rewrite the k-th element of the sum in the right-hand side of Equation (3.31) as

$$\mathbf{D}_k\mathbf{H}_k\mathbf{f} = d_k(x,y)\left(f * h_k\right)(x,y) \tag{3.32}$$

The computation of (3.31) can be easily parallelised, as each $\mathbf{D}_k\mathbf{H}_k\mathbf{f}$ does not depend on any of the other contribution in the summation. Further optimisation is provided by Nagy and O'Leary in [13].

Now, since we have a way to represent the system either via linear algebra manipulation, we can write an estimate of the true image via an optimisation routine like the conjugate gradient method. Furthermore, since we have a direct relation between the matrix-vector multiplications and the convolution via Fourier transform, initial optimisation is straightforward.

Lauer's Space-Variant Deconvolution

Another option for space-variant image restoration is provided by Lauer [14].

Equation (3.3) is rewritten as

$$g(x,y) = \int\int_{\mathbb{R}^2} f(\xi,\eta)h(x,y,x-\xi,y-\eta)d\xi d\eta$$

where g, f and h are the degraded image, the input and the space-variant PSF, respectively. The PSF is rewritten as

$$h(x,y,\xi,\eta) = \sum_{i=1}^{+\infty} a_i(\xi,\eta)h_i(x,y) \tag{3.33}$$

where the $\{h_i\}$ are orthogonal PSF components and each a_i describes the variation of the i-th component of the PSF in the image.

Now, the superposition integral can be rewritten as

$$g(x,y) = \sum_{i=1}^{+\infty}\left(\int\int_{\mathbb{R}^2} f(\xi,\eta)a_i(\xi,\eta)h_i(x-\xi,y-\eta)d\xi d\eta\right) \tag{3.34}$$

which implies that each element in the outer summation is generated by a space-invariant imaging system with PSF h_i. Here $f(\xi,\eta)a_i(\xi,\eta)$ can be interpreted as the input image for the i-th contribution to the summation in (3.34), which implies that the true image $f(\xi,\eta)$ is either weighted or masked by $a_i(\xi, \eta)$ at each location. Furthermore, from Equation (3.34), we see that each PSF h_i is now space-invariant.

To compute each h_i, we use the Karhunen-Loève decomposition (or, sometimes, transform), which is based on the covariance matrix of p PSF observations that are distributed over the image. Let $\tilde{h}_i$ be the i-th observation of the PSF such that each $\tilde{h}_i$ is normalised to a common value, it is centred and all the $\tilde{h}_i$ have the same shape, i.e. they are all $N \times M$ matrices with $N, M \in \mathbb{N}$. Let $\tilde{\mathbf{h}}_i$ be the vector representation of $\tilde{h}_i$.

Let $\mathbf{C}$ be the $p \times p$ covariance matrix generated from the PSF observations. Please note that the covariance matrix is Hermitian and thus is diagonalisable and has real eigenvalues. If the PSFs are real, then the covariance matrix is symmetric.

Let λ_i and $\mathbf{x}_i$ be the eigenvalues and eigenvectors of $\mathbf{C}$. Then, we define the *eigen*-PSF—or the basis functions for the PSF—as

$$\mathbf{h}_i = \sum_{j=1}^{p} x_{i,j} \tilde{\mathbf{h}}_j \tag{3.35}$$

where $x_{i,j}$ is the j-th element of the i-th eigenvector.

Please note that sometimes the PSF may be described by $\tilde{p} < p$ eigen-PSFs, which can significantly increase performance especially when $\tilde{p} \ll p$. The choice of $\tilde{p}$ can be done by sorting the eigenvalues λ_i and assessing their magnitude.

Choices for the model of $a_i(\xi, \eta)$ are driven by the context [14]. Renormalisation of the PSF is also possible.

Each of the $\tilde{p}$ deconvolutions in Equation (3.34) may be performed via Richardson-Lucy's deconvolution as suggested by Lauer [14].

Chapter 4

Computed Tomography

Computed Tomography (CT) is an inverse problem that aims to reconstruct an object from a set of its projections. While it is possible to develop a solution for thee-dimensional objects, we will work in two dimensions for the sake of simplicity. Furthermore, the idea behind it will be explained using a non-diffracting incident parallel beam.

Let $f(x, y)$ be a function that describes a sample, and let $P_\theta(t)$ be the projection of that sample along one direction. Here the subscript θ represents the angle at which the projection has been acquired, which is the angle between the detector line and the coordinate system of the object. Since $f(x, y)$ is two-dimensional, then its projection is one-dimensional. The relation between $f(x, y)$ and $P_\theta(t)$ is given by [15]

$$\begin{aligned} P_\theta(t) &= \int_{(t,\theta)_{line}} f(x, y) ds \\ &= \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} f(x, y)\delta(x \cos\theta + y \sin\theta - t) dx \right) dy \end{aligned} \tag{4.1}$$

where the first equivalence implies that the projection of an object is the path integral through the object and the coordinates (x, y) and (t, s) are related by the rotation matrix in two dimensions,

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.2}$$

where θ is the angle of rotation. The projections can be generated by rotating either the imaging system (as it is done in medical imaging) or the object. For a visual representation please see Fig. 4.1.

For X-ray computed tomography, $P_\theta(t)$ is related to the attenuation of the incident beam within a sample, and $f(x, y)$ represents the mass attenuation coefficient.

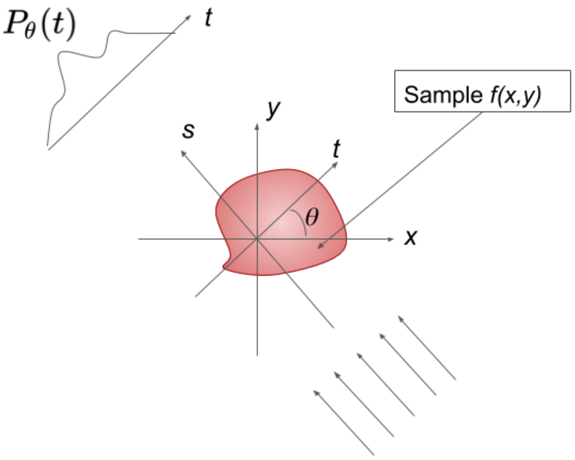


Fig. 4.1: Coordinate system and formation of the projection.

If the refractive index is the quantity of interest (namely, $f(x, y)$), then the projection is the phase accumulated as the light beam travels through the sample, and the refractive index can be calculated from the projected phase.

The set of projections $P_\theta(t)$ is the sinogram of the object. When we have a finite number of projections, the sinogram can be represented by a matrix whose rows represent the projections of the object at different angles. Please note that if the detector is two-dimensional (e.g. an $M \times N$ matrix), each of its rows represents the projection of a different slice.

Let S be the number of projections. Stacking all the images into a three-dimensional $S \times M \times N$ array, we can see that each sinogram can be obtained by reslicing the dataset. Let $A(i, j, k)$ be the three-dimensional array, where i represents the projection number, j the row and k the column of the image. Then, the sinogram for the j-th row is obtained by taking all the elements with fixed j in the array.

Radon Transform of a Circle

The projection of a uniform circle is offered as an example in order to offer some insight on the calculation of the path integral through an object. A uniform circle of radius r is defined by a function $f : \mathbb{R}^2 \to \mathbb{R}$ such that

$$f(x, y) = \begin{cases} \mu & if \quad \sqrt{x^2 + y^2} \leq r \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

or alternatively by $f : \Omega \to \mathbb{R}$ such that $\forall(x, y) \in \Omega\ f(x, y) = \mu$. In this case, the domain of f (namely, Ω) is $\{(x, y) \in \mathbb{R}^2 \mid\ x^2 + y^2 \leq r^2\}$.

Please note that Equation (4.3) is circularly symmetric because Equation (2.25) holds. Furthermore, the radius is constant during any rotation. From this considerations, we can say that the projection of a circle centred in the origin is the same for all angles of rotation. Therefore, we can take $\theta = 0$, which implies $(t, s) = (x, y)$ and $ds = dy$ to simplify the calculation.

Since the $f(x, y)$ is 0 outside of the circle, in those regions its projection will be 0, and the projection for the position x must be calculated in the range from $-r$ to r. By inverting $\sqrt{x^2 + y^2} = r$, the integration over y is restricted to the interval $[-\sqrt{r^2 - x^2}, \sqrt{r^2 - x^2}]$, and we have

$$\begin{aligned} P_\theta(t) = P_0(x) &= \begin{cases} \mu \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} dy & if\ \ -r \leq x \leq r \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 2\mu\sqrt{r^2 - x^2} & if\ \ -r \leq x \leq r \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.4}$$

4.1 Filtered Back Projection

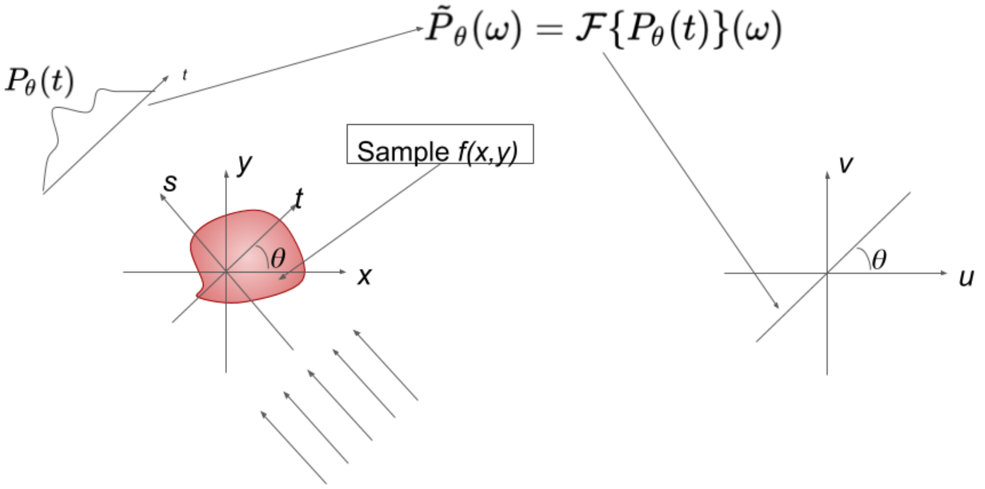

$\tilde{P}_\theta(\omega) = \mathcal{F}\{P_\theta(t)\}(\omega)$
$P_\theta(t)$
t
Sample f(x,y)
s
y
t
θ
x
v
θ
u

Fig. 4.2: Fourier's slice theorem links the two dimensional Fourier transform of the object to the one-dimensional Fourier transform of the sample.

In theory, if an infinite number of positions and projections are available, the inverse Radon transform yields the object itself. However, this is not feasible in practice. Filtered Back Projection (FBP) is an algorithm that aims to reconstruct the object given a set of S projections, each with N pixels and each generated by an incident parallel beam.

The groundwork behind FBP is Fourier's slice theorem, which states that the one-dimensional Fourier transform of the projection of a sample at the angle θ yields the values of a line subtending the angle θ with the u-axis in the two-dimensional Fourier transform $\mathcal{F}f(u,v)$ of the object [15]. A visual explanation of the theorem is shown in Fig. 4.2.

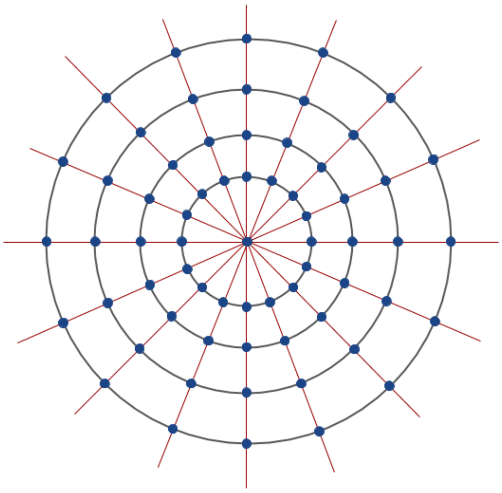


Fig. 4.3: The arcs between two samples became larger as the distance from the centre increases. In the Fourier domain, this means that the high frequencies are under-sampled. Reconstruction without filtering yield severe blurring, hence the name Filtered Back Projection.

However, it is still not enough to obtain a good estimate of the object. Indeed, if we simply collected all the lines and put them together in the Fourier domain, we would have a situation similar to Fig. 4.3. This means that the dimension of the arc that must be interpolated in the frequency domain increases with the radius r. This dependence is linear. Indeed, for a fixed angular step $\Delta\theta$, the length a of an arc of a circle of radius r is given by

$$a = \Delta\theta r \tag{4.5}$$

The introduction of a filter helps to fix this problem. This operation can be seen as weighting the frequencies, which helps reduce errors [15]. Traditionally, the kernel used to filter the projections, is the Ram-Lak filter, which is best known as ramp filter due to its shape in the Fourier domain,

$$k(\omega) = \begin{cases} |\omega| & if \;\; |\omega| \leq \frac{1}{2\tau} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where ω is the frequency and τ is the sampling interval—namely, the one-dimensional pixel size.

Such a kernel tends to favour the high frequency components of the Fourier transform of the object and to diminish the importance of the low frequencies. This improves the edges as they are related to the high frequency content of the Fourier transform. However, this part of the spectrum is often associated with noise as well. Therefore, the application of the ramp filter may worsen the quality of noisy images. To fix this, there are two strategies: the filtered Fourier transform could be smoothed with a windowing function (e.g. the Hamming window) or the projection may be filtered with a softer kernel, which initially behaves roughly like a ramp, reaches a peak for a certain frequency and slowly decreases to diminish the importance of the higher frequencies or to cut them off. Please note, however, that the employment of either solutions smooths the image and thus introduces additional blurring which worsens spatial resolution.

The reconstruction of the object from its projections can be performed either in the spatial or in the frequency domain, however it is more accurate to interpolate in the former [15]. Interpolation is usually necessary since it is likely that the projection of the point through (4.2) does not lie to a measured sample on the detector line. However, linear interpolation is usually sufficient.

Let k be the chosen kernel (e.g. the ramp filter) and w the smoothing window. For any projection $P_\theta(t)$ acquired at an angle θ, we define the filtered projection $Q_\theta(t)$ as

$$Q_\theta(t) = \mathcal{F}^{-1}\{\mathcal{F}\{P_\theta\}\mathcal{F}\{k\}w\}(t) \tag{4.7}$$

Please remember that w is optional and leaving it out is equivalent to defining $w(\omega) = 1$ for all the frequencies.

Equation (4.7) can be further simplified to $Q_\theta(t) = (P_\theta * \tilde{k})(t)$ if we define the kernel as $\tilde{k} = \mathcal{F}^{-1}\{\mathcal{F}\{k\}w\}$.

Given the filtered projection $Q_\theta(t)$, the object $f(x, y)$ is defined as

$$\begin{aligned} f(x,y) &= \int_0^\pi Q_\theta\left(t(\theta)\right) d\theta \\ &= \int_0^\pi Q_\theta(x\cos\theta + y\sin\theta) d\theta \end{aligned} \tag{4.8}$$

where Equation (4.2) has been used. The integration is carried out only from 0 to π because the projections generated from π to 2π do not yield new information, as they can be obtained by mirroring the projections with respect to the centre of rotation.

The discrete version of Equation (4.8) is

$$f(x, y) = \frac{\pi}{S} \sum_{i=0}^{S-1} Q_{\theta_i}(x \cos \theta_i + y \sin \theta_i) \tag{4.9}$$

where S is the number of projections, the origin of the coordinate system is at the centre of the slice (centre of the reconstruction), and x and y are the sampled coordinates such that for any $m, n = 0, 1, ..., N - 1$ we have

$$x = m - c \qquad y = c - n$$

Here (c, c) is the centre of the slice. Please note that the x and y coordinates of the centre are identical because the reconstructed slice is a square of side N. Furthermore, if we have a constant angular step, then we can rewrite the i-th angle of acquisition as $\theta_i = i\Delta\theta = \frac{i\phi}{S}$ where ϕ is the whole angular interval. For example, if we acquire 1800 projection in the range form 0 to π, then $\Delta\theta = 0.1^\circ$ ($\approx 1.745 \cdot 10^{-3}$ rad).

Thus, the FBP algorithm for the spatial domain can be summarised into the following steps

Step 1: filter all the projections with a kernel k and apply the smoothing window if needed;

Step 2: for each pixel, sum the projections over all angles according to Equation (4.9). This step usually requires interpolation. Please note that the multiplication by the constant normalisation factor can be done after the summation step as it is suggested by Equation (4.9).

The algorithm can be easily implemented using parallel computing because all the projections in the sinogram can be filtered independently and all the pixels of the reconstructed slice can be computed in parallel.

Furthermore, we define the circle of reconstruction as that circle for which we have $\sqrt{x^2 + y^2} \leq \tau \frac{N}{2}$ provided that the origin is at the centre of the reconstructed slice and that τ is the pixel size in the detector line. Values outside of the circle of reconstruction will be missing in some projection and, hence, cannot be accurately reconstructed. Therefore, they are often ignored during FBP, which speeds up the computation.

4.1.1 Practical Considerations for a CT System

From Equations (4.2) and (4.9), we see that the object rotates about an axis or that, alternately, the system rotates about the object. However, it must be stressed that the axis of rotation must be orthogonal to the incident parallel beam. If either were tilted, then the projections on the detector row would not represent a single slice and points from the object would move onto different lines, as shown in Fig. 4.4. In other words, we can say that the projections at two angles of acquisition are generated from two different objects when the rotation axis is tilted. Therefore, alignment of the system is of the utmost importance.

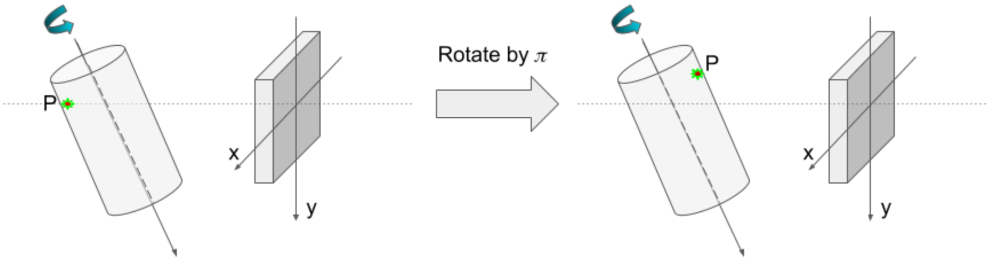


Fig. 4.4: When the rotation axis is tilted with respect to incident parallel beam, the projections of the object at two different angles does not come from the same slice.

On the other hand, it is not required that the centre of rotation lies at the centre of the detector width as it can be retrieved via human inspection or algorithms [16,17]. This displacement can be removed by adding or removing enough columns in the sinogram, thus changing the number of samples along the detector line. It is also possible to introduce the displacement in the reconstruction algorithm following a similar approach to the one used for fan beam FBP by Gullberg *et al.* in [18].

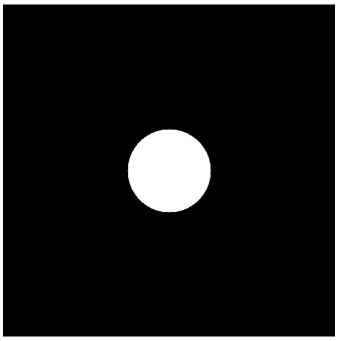
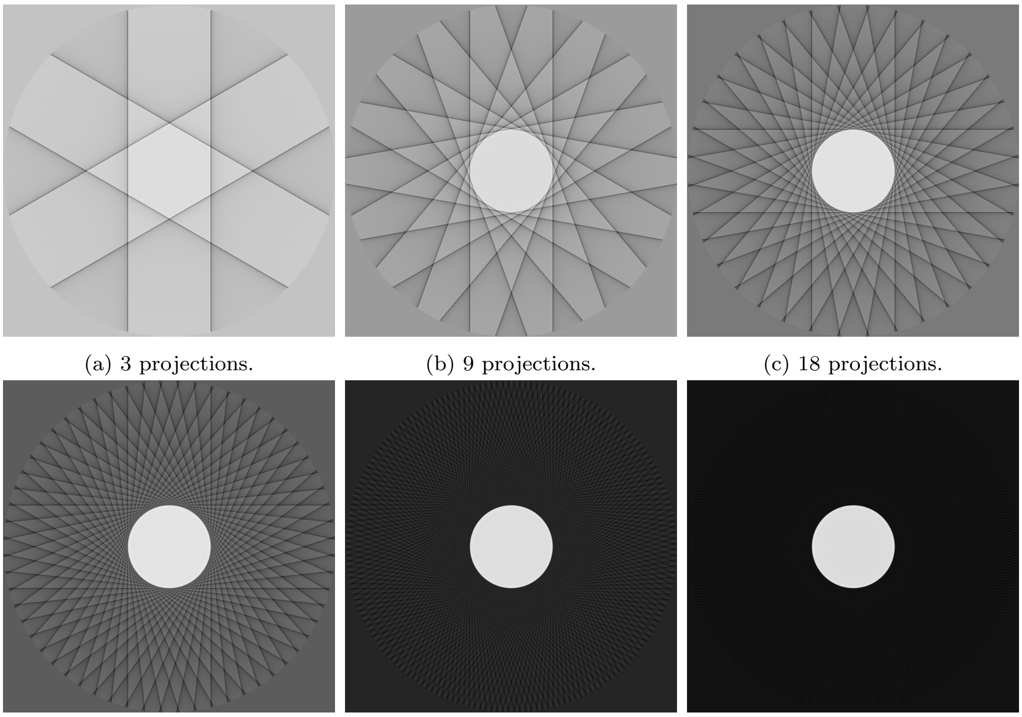


Fig. 4.5: Ideal circle from Equation (4.3). Here, $r = 64$, $\mu = 4$. The image is a square of side $N = 512$.

Ideally, the higher the number of projections the better the reconstructed slice is. However, this is often impractical due to the large dimension of the datasets, the difficulty to acquire images at different angles or the rotation stage limits. A good rule of thumb is that the number of projections and the samples on the detector line should be of the same order of magnitude.

As an example, we offer the reconstruction of a uniform, noise-free circle from Equation (4.3), which is show in Fig. 4.5. The number of pixels on the detector row was set to 512, and the projection were generated by stacking (4.4) into a sinogram. Please note that this is possible because the projection for a circle centred in the centre of reconstruction is the same for all angles and therefore we only need to compute it once.

The circle was reconstructed with 3, 9, 18, 30, 90 and 180 projections, and the results are shown in Fig. 4.6.



(a) 3 projections.

(b) 9 projections.

(c) 18 projections.

(d) 30 projections.

(e) 90 projections.

(f) 180 projections.

Fig. 4.6: Reconstruction of a circle with different number of projections.

Please note that if the number of projections is too low the circle cannot be properly reconstructed. While the object can be guessed from as low as 9 views, the edges are still approximated and the radial lines in the background pattern are still evident. Increasing the number of projections greatly improves the situation and reduces this undesired artefacts.

4.2 Diffraction Tomography

The basis for diffraction tomography is Fourier's diffraction theorem. Like Fourier's slice theorem, it links the Fourier transform of a projection to the Fourier transform of the object itself. However, in this case the Fourier transform of the projection acquired by illuminating a weakly diffracting object with an incident plane wave at an angle θ is a semicircular arc that passes through the origin in the Fourier domain and has the line which subtends θ with the u-axis as a tangent [19]. Please note that it is recommended to acquire the projections from all angles and not limit the acquisition to the half-plane (i.e. acquiring only from the angle interval $[0, \pi]$).

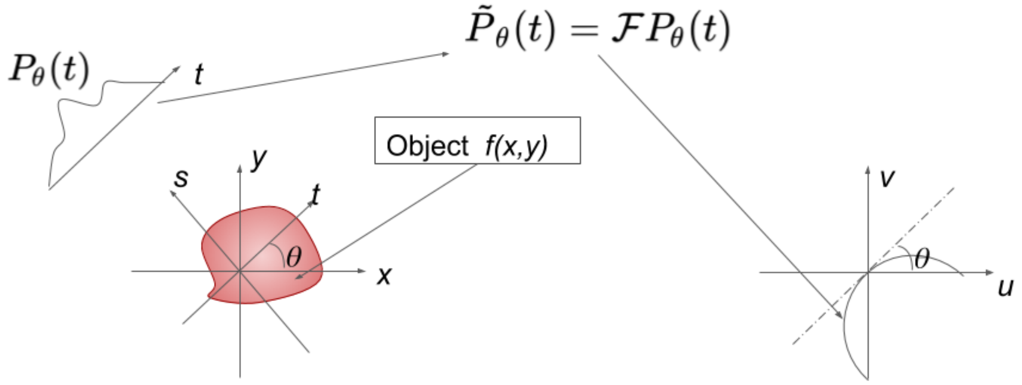


Fig. 4.7: Fourier's diffraction theorem.

One approach for the reconstruction of the object from the diffracted waves is the interpolation in the Fourier domain. However, first the Fourier transform of the projection must be weighted by applying either the first Born approximation or the first Rytov approximation. Then, the coordinates of the point on the arc are retrieved, and all the point on the two-dimensional grid in the Fourier domain are computed via interpolation. Indeed, usually the samples on the arc might not lie exactly at the position at which the grid is sampled. Furthermore, interpolation must be used to retrieve the missing frequencies on the grid. The object is found by taking the two-dimensional inverse Fourier transform [19, 20].

Chapter 5

Light Waves

In physics, light can be described either with massless particles called photons or with wave theory, depending on the application. The former model is often used for the modelling of X-rays and γ-rays, while the latter can be employed to describe the behaviour of visible light. For this reason, the electromagnetic (EM) waves in the visible spectrum are also sometimes called light waves. Furthermore, in this case it is common to consider only the electric component in the treatment.

A wave is a solution to the partial differential equation

$$\nabla^2 \psi - \frac{1}{v^2} \frac{\partial^2 \psi}{\partial t^2} = 0 \tag{5.1}$$

where ψ is the wave, v is its speed, and ∇^2 is the Laplacian, which is defined as

$$\Delta = \nabla^2 = \sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} \tag{5.2}$$

for a function with N independent variables. For the wave equation, the Laplacian is used on the spatial coordinates.

Using Maxwell's equations for the EM field in the vacuum, we can find a similar expression for the electric field,

$$\nabla^2 \mathbf{E} - \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = 0 \tag{5.3}$$

where $\mathbf{E}$ is a three-dimensional vector representing the electric field and $c = \frac{1}{\sqrt{\epsilon_0 \mu_0}}$ is the speed of light in the vacuum. The constants ϵ_0 and μ_0 are permittivity and permeability of free space.

$\mathbf{E}$ vibrates perpendicularly to the direction of propagation. In many application, however, its vibrations are limited to a single line and always in a plane perpendicular to the propagation direction, in which case the EM wave is said to be *linearly polarised* and can be treated with a scalar wave equation,

$$\nabla^2 E - \frac{1}{c^2}\frac{\partial^2 E}{\partial t^2} = 0 \tag{5.4}$$

It can be verified that the plane wave

$$E(\mathbf{r};t) = E_0 e^{i(\mathbf{k}\cdot\mathbf{r} - \omega t + \phi_0)} \tag{5.5}$$

is a solution to the wave equation in (5.4). In Equation (5.5), $E_0 \in \mathbb{R}$ is the amplitude of the wave, and the argument of the exponential is called phase. Please note, however, that, for certain processes, the phase will be defined by

$$\phi = \mathbf{k}\cdot\mathbf{r} + \phi_0 \tag{5.6}$$

The vector quantity $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$ is the set of coordinates, ϕ_0 is the initial phase (or phase constant), ω is the angular frequency, and $\mathbf{k} = (k_x, k_y, k_z)$ is the wave vector, whose modulus is the wave number k

$$|\mathbf{k}| = k = \frac{2\pi}{\lambda} \tag{5.7}$$

where λ is the wavelength.

Furthermore, ω is related to the frequency ν of the light wave by

$$\omega = 2\pi\nu \tag{5.8}$$

For any EM wave in free space, we have

$$\lambda\nu = c \tag{5.9}$$

which means that we can determine the wavelength if the frequency is known, and vice versa.

Alternately, we can use the opposite sign convention and write the plane wave as

$$E(\mathbf{r};t) = E_0 e^{i(-\mathbf{k}\cdot\mathbf{r} + \omega t - \phi_0)} \tag{5.10}$$

and change the sign to the phase definition in Equation (5.6). This is the convention used by Schnars in [21]. In general, all the results and considerations made for one sign convention can be adapted to the other with little difficulty.

Many books of Fourier optics and holography [2,22,23] use the convention in (5.5).

Please note that it is usually possible to change the coordinate system in order to have the direction of propagation parallel to the z-axis and write $E(z;t) = E_0 e^{i(kz-\omega t+\phi_0)}$.

Since detectors usually measure the intensity, we introduce the intensity for any EM wave as

$$I = |E|^2 = E\overline{E} \tag{5.11}$$

5.1 Electromagnetic Spectrum

It is easy to see that there are an infinite number of pairs λ, ν for which Equation (5.9) holds. The term electromagnetic spectrum is used to describe the range of frequencies and wavelengths covered by EM radiation and to compartmentalise it according to properties, use or method of production. An example is shown in Fig. 5.1.

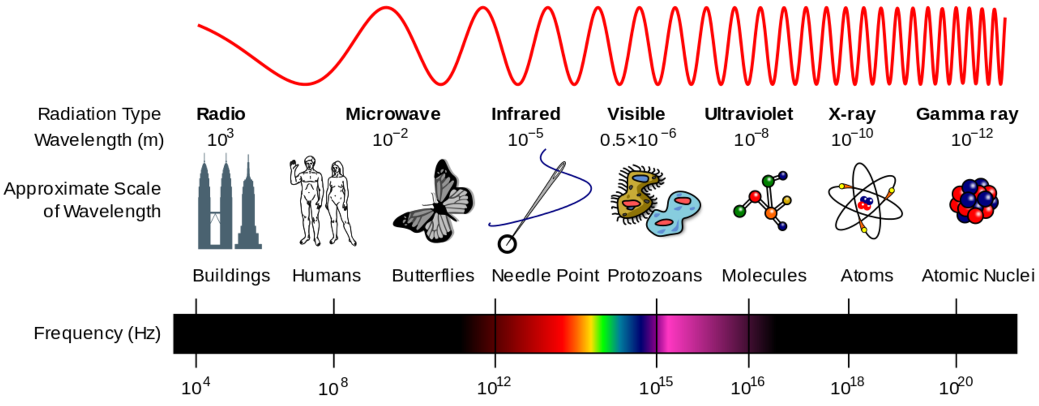


Fig. 5.1: Electromagnetic spectrum.

In particular, we distinguish

- radio waves—which are most notably used in radios and televisions;
- microwaves—which are employed in microwave ovens, Wi-Fi and radars;
- infrared (IR)—which is just below the visible light and is employed in night and thermal vision;
- visible light—which comprises EM waves with wavelength between about 700 nm (red) and about 400 nm (violet). It is the only range of the EM spectrum that is visible to the human eye. EM waves in this range are called light waves;

- ultraviolet (UV)—which is responsible for sunburn. It is also the first region of the EM spectrum for which we have ionising radiation, which is harmful to humans. Indeed radiation becomes harmful when the energy of the photons is greater than 10 eV. Please note that photon energy is defined as $J = h\nu$, where $h = 6.626 \cdot 10^{-34}$ m^{2}kgs^{-1} is the Planck constant;
- X-rays—X-rays have great penetrating power that makes them useful for diagnostic, non-invasive medical imaging (e.g. radiography and CT) and therapy but are a type of ionising radiation and thus harmful to biological tissues. They are generated from the interaction of electron or photons with atoms;
- γ-rays—γ-rays are also used in medical imaging (e.g. SPECT and PET). The X- and γ-ray energy ranges partially overlap, and the only difference between these two types of radiation lies in their origin. Indeed, γ-rays are generated by radioactive decay: the nucleus release a γ photon to de-excite.

5.2 Interference

Let us consider a set of N waves, where N is a positive integer such that $N \geq 2$. Let $\mathbf{E}_j(\mathbf{r}; t)$ be the solution to the wave equation (5.3) for the j-th wave in the set. The spatial superposition

$$\mathbf{E}(\mathbf{r}; t) = \sum_{j=1}^{N} \mathbf{E}_j(\mathbf{r}; t) \tag{5.12}$$

is called interference and is a solution to the wave equation itself since a linear combination of solutions to (5.3) is also a solution.

For the sake of simplicity, let E_1 and E_2 be two monochromatic waves that are solutions to Equation (5.4). Let $E_n = a_n e^{i\phi_n}$ with $a_n, \phi_n \in \mathbb{R}$, $n = 1, 2$. We can compute the resulting complex amplitude and intensity with Equations (5.12) and (5.11) respectively,

$$\begin{aligned} I &= (E_1 + E_2)\overline{(E_1 + E_2)} \\ &= a_1^2 + a_2^2 + a_1 a_2 \left(e^{i(\phi_1 - \phi_2)} + e^{-i(\phi_1 - \phi_2)}\right) \\ &= a_1^2 + a_2^2 + 2a_1 a_2 \cos\left(\phi_1 - \phi_2\right) \end{aligned} \tag{5.13}$$

where the second equivalence is obtained through straightforward algebraic manipulations and the last holds because the cosine is even and the sine is odd. Indeed, from Euler's formula, we have

$$\begin{aligned} e^{i\phi} + e^{-i\phi} &= \cos\phi + i\sin\phi + \cos(-\phi) + i\sin(-\phi) \\ &= \cos\phi + i\sin\phi + \cos\phi - i\sin\phi \\ &= 2\cos\phi \end{aligned} \tag{5.14}$$

Since the cosine oscillates between -1 and 1 with a period of 2π, and since $\cos 0 = 1$ and $\cos\pi = -1$, we see that we have a maximum if $\Delta\phi = \phi_1 - \phi_2$ is an even integer multiple of π and a minimum if it is an odd integer multiple of π.

When the intensity I reaches a maximum, we will say that we have a constructive interference and that the condition for this case is

$$\Delta\phi = 2m\pi \quad m \in \mathbb{Z} \tag{5.15}$$

On the other hand, when I reaches a minimum, we will say that we have a destructive interference, which happens when

$$\Delta\phi = (2m+1)\pi \quad m \in \mathbb{Z} \tag{5.16}$$

On a detector, constructive and destructive interference appears as a series of bright and dark lines—or *fringes*—respectively and forms the *interference pattern* [21].

Please note that if $a_1 = a_2$, then the minimum intensity will be 0 and the maximum intensity will be $4a_1^2$.

5.3 Spatial and Temporal Coherence

Real sources are, in general, extended and not monochromatic, which poses limitations on any experiment based on interference. The extension of a source is related to the *spatial coherence* of a wave, while the existence of non-monochromatic waves is related to its *temporal coherence.*

5.3.1 Spatial Coherence

Spatial coherence is the degree of correlation between two spatially distinct points of the same wave field. It also describes the ability of a wave to interfere.

Let us consider the system in Fig. 5.2, which describes Young's interference experiment with two slits a_1 and a_2 on and an extended source S of size h. Please note that the source need not be placed on axis, but this placement simplifies the explanation and does not incur in loss of generality.

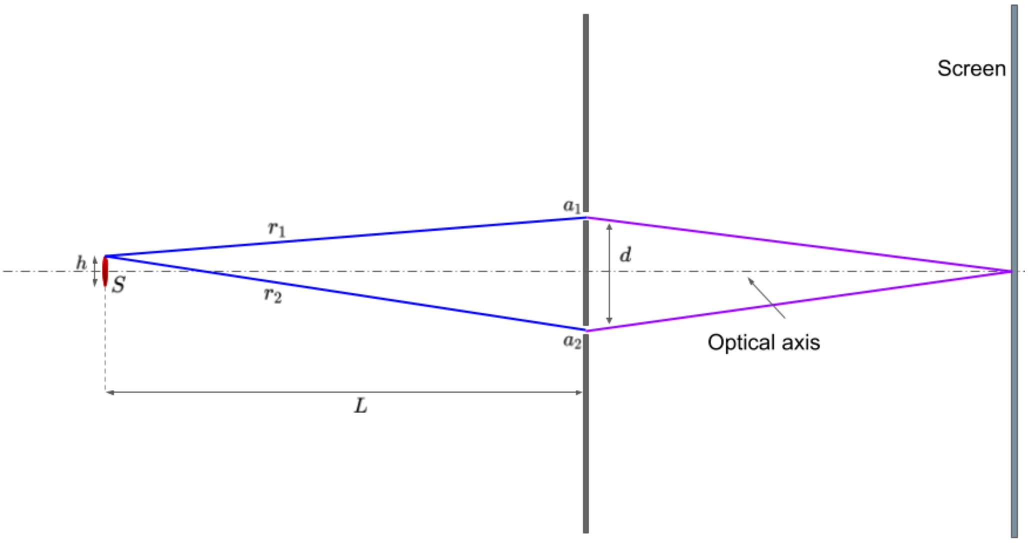


Fig. 5.2: Set-up for Young's experiment with an extended source S.

An extended source can be considered as a superposition of point sources located at different positions within S. From the figure, it is clear that $r_1 \neq r_2$ unless we consider an on-axis point source, which implies that the phases ϕ_1 and ϕ_2—respectively at the location a_1 and a_2—will generally be different. The interaction of the wave generated by the source S with the slits a_1 and a_2 will generate two spherical waves, which will interfere on the screen, creating a fringe pattern. The fringe visiblity, which describes the degree of spatial coherence between the sources located at a_1 and a_2, is defined as [24]

$$\gamma = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{5.17}$$

where I_{max} and I_{min} are, respectively, the maximum and minimum intensities detected on the screen. Please note that here γ is exactly the Michelson contrast and is defined in the interval $[0, 1]$. If $\gamma = 0$, all the contrast on the screen is lost, which, in this case, means that no fringe is visible. If $\gamma = 1$, then the minimum intensity is 0 and $I_{min} \neq I_{max}$.

To improve the description, let us first consider an extended source that lies entirely on the optical axis (i.e. a segment). In this case, ϕ_1 and ϕ_2 are the same for all the points in the source. This means that extending the source along the axis does not change the phase relation between a_1 and a_2.

On the other hand, we exploit the geometry of the system to describe the phase difference between two points on an extended source perpendicular to the optical axis. If d is small, we can describe the path difference as $r_2 - r_1 \approx d \sin\theta$, where θ is the angle shown in Fig. 5.3. Here, we can put

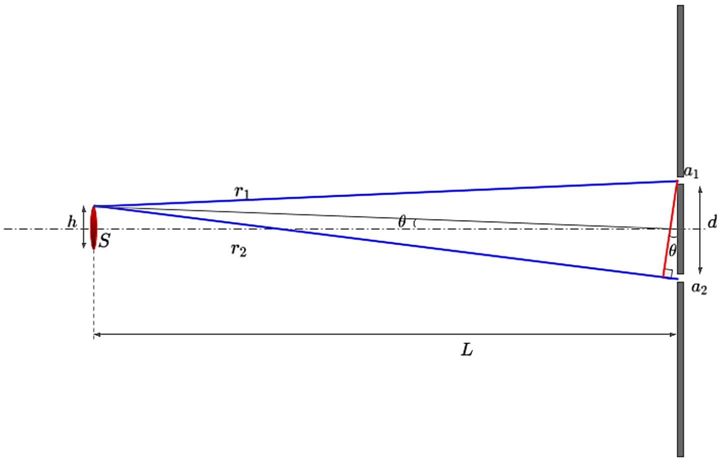


Fig. 5.3: Geometry of the system in Fig. 5.2.

the centre of the source on the optical axis. Furthermore, since L is usually large, we can write $\sin\theta \approx \frac{h}{2}\frac{1}{L} = \frac{h}{2L}$.

Since there is a linear relation of the type $\phi = kr$ between the path length and the phase, we can write

$$\phi_2 - \phi_1 = kd\sin\theta = k\frac{dh}{2L} = \frac{\pi dh}{\lambda L} \tag{5.18}$$

where the last equivalence holds because k is the wavenumber.

Now, the limiting source size h can be calculated by considering that the two ends of the segment perpendicular to the optical axis will produce contributions that cancel out if S is too large, which implies that there is a certain h_{lim} that describes the limiting size of the source, which is defined by

$$(\phi_2 - \phi_1)_u - (\phi_2 - \phi_1)_d = \pi$$

where the subscripts u and d are used for the point-source above and below the optical axis, respectively [24]. Therefore, we can write h_{lim} as

$$h_{lim} = \frac{\lambda L}{2d} \tag{5.19}$$

and introduce the *coherence distance* d_k through the relation [21]

$$\frac{d_k h}{2L} = \frac{\lambda}{2} \tag{5.20}$$

Therefore, the spatial coherence depends on the wavelength of the impinging light wave and on the interferometer geometry. In particular, we can say that it decreases with the wavelength and that h_{lim} and d_k are inversely proportional. Experiments based on interference should not be attempted without first ensuring that the system follows the spatial coherence condition [24]

$$\frac{\lambda L}{hd} \leq 2 \tag{5.21}$$

which can be found by considering that if h is suitable, then $h \leq h_{lim}$.

5.3.2 Temporal Coherence

As mentioned above, real sources are not perfectly monochromatic. We define the temporal coherence as a measure of the correlation between a light wave and a time-delayed copy of itself. Let us suppose that we have a source that emits light with wavelength $\lambda \pm \Delta\lambda$. Let two waves with wavelength λ and $\lambda + \Delta\lambda$ interfere constructively. There is a critical length L, called *coherence length*, above which this no longer happens. We can now introduce the *coherence time* τ [21],

$$\tau = \frac{L}{c} \tag{5.22}$$

which is the time interval in which the wave propagates.

5.4 Diffraction

In optics, when the quantity of interest is the spatial distribution of the wave, it is generally accepted to drop the temporal component in Equation (5.5) and write

$$E(r) = e^{i(kr+\phi_0)} \tag{5.23}$$

For this reason, Equation (5.23) will be used hereafter unless otherwise stated.

Now, let us consider a plane wave impinging on an opaque screen with one or more holes, like the one in Fig. 5.4, or—vice versa—a transparent screen with some opaque obstacles.

If the size of the obstacle and the wavelength have the same order of magnitude, then the interaction between the impinging plane wave and the obstacle will generate a pattern of alternating dark and bright regions called *diffraction*. We will also say that the wave is diffracted by the obstacle.

Diffraction is described by Huygens' principle, which states that we can consider every point of a wave front as a source of (secondary) spherical waves and that we can obtain the wave field through the superposition of all the secondary spherical waves, as shown in Fig. 5.5.

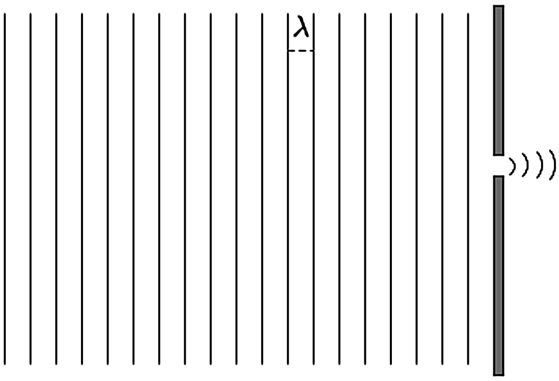


Fig. 5.4: A plane wave impinging on an opaque screen with a hole.

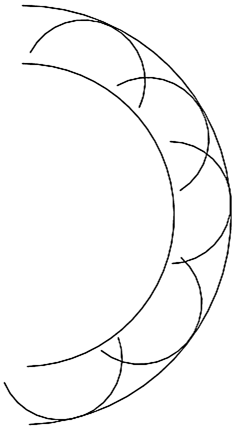


Fig. 5.5: Each point of the leftmost wave front can be seen as a source of secondary spherical waves. Their superposition generates the wave field.

Therefore, each point within the hole in Fig. 5.4 can be seen as a source of spherical waves.

The mathematical behaviour of diffraction is described by the Fresnel-Kirchhoff integral,

$$U_1(x,y) = -\frac{i}{\lambda}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} U(\xi,\eta)\frac{e^{ik\rho'}}{\rho'}Q(\theta,\theta')d\xi d\eta \tag{5.24}$$

where λ is the wavelength of the impinging light wave, $U(\xi,\eta)$ is the wave field at the aperture (or obstacle) plane and k is the wavenumber. To explain the meaning of $Q(\theta,\theta')$ and ρ', we first have to introduce the geometry of the system in Fig. 5.6.

The coordinates in the source plane are not important here, so they have

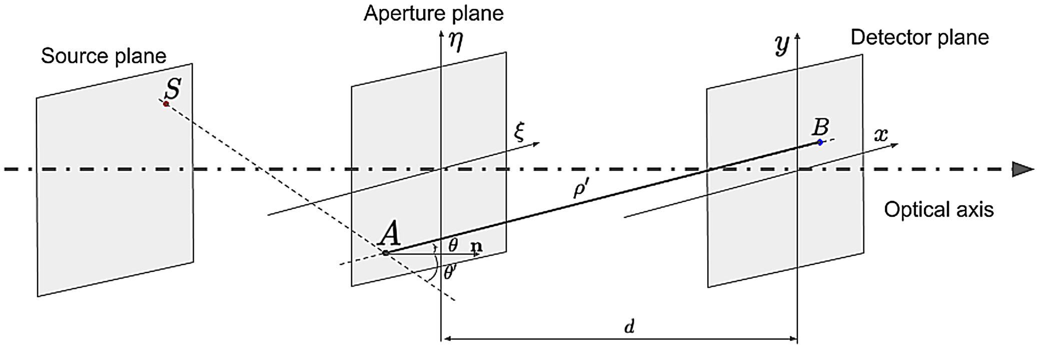


Fig. 5.6: Geometry of the acquisition system for a diffraction experiment.

been dropped for readability. In Fig. 5.6, S is the source, d is the recording—or, sometimes, reconstruction—distance, which is the distance between the aperture and the detector planes. The distance between the hole (or obstacle) and any point of the detector plane is ρ' and can be calculated as

$$\rho' = \sqrt{(\xi - x)^2 + (\eta - y)^2 + d^2} \tag{5.25}$$

The angles θ' and θ are the angle that the straight lines passing through SA and AB form with the normal $\mathbf{n}$ to the aperture plane. $Q(\theta, \theta')$ is the inclination factor and is used to exclude propagation in the backward direction since it does not happen experimentally [21]. This factor is written as

$$Q(\theta, \theta') = \frac{1}{2}\left(\cos\theta + \cos\theta'\right) \tag{5.26}$$

Please note that in most practical cases $\theta, \theta' \approx 0$, and therefore $Q(\theta, \theta') \approx 1$.

There are other expression for scalar diffraction, for example the two Rayleigh-Sommerfeld diffraction solutions [2],

$$U_1(x, y) = -\frac{i}{\lambda}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} U(\xi, \eta)\frac{e^{ik\rho'}}{\rho'}\cos\theta d\xi d\eta \tag{5.27a}$$

$$U_1(x, y) = -\frac{i}{\lambda}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} U(\xi, \eta)\frac{e^{ik\rho'}}{\rho'}d\xi d\eta \tag{5.27b}$$

which are valid for an infinitely distant point source that therefore produces normally impinging plane waves onto the aperture plane. Please note that under this hypothesis, the inclination factor (5.26) for the Fresnel-Kirchhoff integral in Equation (5.24) becomes

$$Q(\theta, \theta') = \frac{1}{2}(1 + \cos\theta) \tag{5.28}$$

In this case, the Fresnel-Kirchhoff integral is an average of the two Rayleigh-Sommerfeld solution [2].

Please note that different sign conventions leads to slightly different expressions as in [22] and [21]. In the following section, the expressions for both sign conventions will be presented.

5.4.1 Numerical Diffraction

Equation (5.24) is hard to evaluate directly even with the aid of a computer.

It is, however, possible, to approximate or rewrite the diffraction equation in order to exploit existing fast routines for simulation or computation.

The Fresnel Transform

Let us consider the first Rayleigh-Sommerfeld solution in (5.27a). Alternately, we can consider a source for which $Q(\theta, \theta') \approx 1$ since it is valid in most practical cases or the second Rayleigh-Sommerfeld solution (5.27b).

First, the cosine in (5.27a) can be rewritten as

$$\cos\theta = \frac{d}{\rho'} \tag{5.29}$$

for the geometry in Fig. 5.6.

For the calculation below, we will work under the hypothesis that the paraxial approximation is valid. The paraxial approximation states that the maximum angle generated from the hole (or obstacle) and the detector is small. In other words, $(x - \xi)$ and $(y - \eta)$ must be small with respect to d.

If $x, y, \xi, \eta \ll d$, Equation (5.25) can be expanded in a Taylor series,

$$\rho' \approx d + \frac{(\xi - x)^2 + (\eta - x)^2}{2d} + \frac{1}{8}\frac{\left((\xi - x)^2 + (\eta - y)^2\right)^2}{d^3} + ... \tag{5.30}$$

and the last explicit term in Equation (5.30) can be neglected provided that it is much smaller than the wavelength, or

$$\frac{1}{8}\frac{\left((\xi - x)^2 + (\eta - y)^2\right)^2}{d^3} \ll \lambda \tag{5.31}$$

In this case, we can define a distance ρ as

$$\rho = d + \frac{(\xi - x)^2 + (\eta - y)^2}{2d} \tag{5.32}$$

We will also assume that we can approximate the ρ' in the denominator with d [2].

Direct substitution into the diffraction equation leads to

$$U_1(x,y) = -\frac{i}{\lambda d} e^{ikd} \int\int_{\mathbb{R}^2} U(\xi,\eta) e^{i\frac{k}{2d}\left((\xi-x)^2+(\eta-y)^2\right)} d\xi d\eta \tag{5.33}$$

which is a two-dimensional convolution. Further optimisation can be achieved by explicitly computing the quadratic term,

$$\begin{aligned} U_1(x,y) = &-\frac{i}{\lambda d} e^{ikd} e^{i\frac{k}{2d}\left(x^2+y^2\right)} \\ &\times \int\int_{\mathbb{R}^2} U(\xi,\eta) e^{i\frac{k}{2d}\left(\xi^2+\eta^2\right)} e^{-i\frac{k}{d}x\xi - i\frac{k}{d}y\eta} d\xi d\eta \end{aligned} \tag{5.34}$$

Equation (5.34) is called the Fresnel transform due to its relation to the Fourier transform.

Before delving further into this relation, we first introduce $E(\xi,\eta) = \overline{U}(\xi,\eta)$ and do the conjugate of U_1,

$$\begin{aligned} O(x,y) = &\frac{i}{\lambda d} e^{-ikd} e^{-i\frac{k}{2d}(x^2+y^2)} \\ &\times \int\int_{\mathbb{R}^2} E(\xi,\eta) e^{-i\frac{k}{2d}\left(\xi^2+\eta^2\right)} e^{i\frac{k}{d}(x\xi+y\eta)} d\xi d\eta \end{aligned} \tag{5.35}$$

which is the result in [21].

Temporarily neglecting all the terms outside of the integral in Equation (5.34), we see that the field $U(\xi,\eta)$ is first multiplied by a quadratic phase and then Fourier transformed with the frequencies

$$u = \frac{x}{\lambda d} \qquad v = \frac{y}{\lambda d} \tag{5.36}$$

where the identity in Equation (5.7) has been exploited to match the integral in (5.34) with the 2D Fourier transform in (2.21).

The Fresnel transform can either be computed via a 2D Fourier transform or its inverse depending on whether we are using the + or − sign convention for the phase generated from the spatial components.

The detector (or the screen) usually records the intensity of the diffracted wave, or $|U_1(x,y)|^2$. From Equation (5.34), we see that the phase terms outside of the integral can be dropped and we have

$$|U_1(x,y)|^2 = \frac{1}{\lambda^2 d^2} \left| \mathcal{F}\left\{ U(\xi,\eta) e^{i\frac{k}{2d}(\xi^2+\eta^2)} \right\} \right|^2 \tag{5.37}$$

Fraunhofer Diffraction

The Fraunhofer diffraction is a further approximation of the Fresnel transform for which the quadratic phase within the integral is very small, namely [2, 22, 23]

$$d \gg \frac{k\,(\xi^2+\eta^2)_{max}}{2} \tag{5.38}$$

where the subscript *max* is here used to indicate the maximum possible value of $\xi^2+\eta^2$, which is related to the dimension of the aperture. We will call the approximation in (5.38) the *Fraunhofer approximation.*

Please note that the this approximation requires a large recording distance with respect to the dimension of the aperture and is thus valid only in the *far field.*

The diffraction equation becomes

$$U_1(x,y) = -\frac{i}{\lambda d} e^{ikd} e^{i\frac{k}{2d}(x^2+y^2)} \int\int_{\mathbb{R}^2} U(\xi,\eta) e^{-i\left(\frac{kx}{d}\xi+\frac{ky}{d}\eta\right)} d\xi d\eta \tag{5.39}$$

Following the same considerations previously made for the Fresnel transform, we can rewrite U_1 for the other sign convention as

$$O(x,y) = \frac{i}{\lambda d} e^{-ikd} e^{-i\frac{k}{2d}(x^2+y^2)} \int\int_{\mathbb{R}^2} E(\xi,\eta) e^{i\left(\frac{kx}{d}\xi+\frac{ky}{d}\eta\right)} d\xi d\eta \tag{5.40}$$

Angular Spectrum Method

From the first Rayleigh-Sommerfeld solution (5.27a) and the expressions for ρ' (5.25) and $\cos\theta$ (5.29), we see that we can rewrite the solution as a convolution,

$$U_1(x,y) = \mathcal{F}^{-1}\left\{\mathcal{F}U\mathcal{F}h\right\}(x,y) \tag{5.41}$$

where the kernel h is defined as [22, 23]

$$h(\xi,\eta;d) = -\frac{id}{\lambda}\frac{e^{ikr}}{r^2} \tag{5.42}$$

with $r=\sqrt{d^2+\xi^2+\eta^2}$.

Further optimisation is possible because, for any d sufficiently distant from the aperture, the analytical expression of $H(u,v)=\mathcal{F}h(x,y)$ is known [22] and is

$$H(u,v;d) = \begin{cases} e^{ikd\sqrt{1-\lambda^2(u^2+v^2)}} & if\ \sqrt{u^2+v^2} \leq \frac{1}{\lambda} \\ 0 & \text{otherwise} \end{cases} \tag{5.43}$$

We see that the only requirement introduced here not to filter the Fourier domain with a low-pass filter is that $\sqrt{u^2+v^2} \leq \lambda^{-1}$ where u and v are the spatial frequencies.

The transfer function in Equation (5.43) is sometimes rewritten as [2]

$$H(u,v;d) = e^{ikd\sqrt{1-\lambda^2(u^2+v^2)}}\text{circ}\left(\lambda\sqrt{u^2+v^2}\right) \tag{5.44}$$

where circ(r) is defined as [22]

$$\text{circ}(r) = \begin{cases} 1 & if\ r \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.45}$$

Therefore, we can write

$$U_1(x,y) = \mathcal{F}^{-1}\left\{\mathcal{F}\{U\}H\right\}(x,y) \tag{5.46}$$

To obtain the result for the other sign convention (i.e. the one used by Schnars [21]), we have to change the sign of the phase of U and H.

This numerical diffraction approach is known as *angular spectrum method* or, sometimes, convolution approach [21].

5.5 Lenses in Coherent Imaging

We define a lens as a piece of dense material with refractive index (RI) $n > n_0$, where n_0 is the RI of the surrounding medium, which is usually air. Usually, lenses are made of glass and $n \approx 1.5$. In this section, we will assume $n_0 = 1$ as this is valid in most practical cases.

A lens is considered *thin* if the change of direction introduced by the passage of light through it is negligible. In this case, it simply produces a phase delay due to the difference in RI.

As an example, we present a lens in Fig. 5.7. If it is thin, we expect that the phase delay due to the lens material and thickness will be $kn\Delta(x,y)$. However, the lens does not have uniform thickness. We can write the total phase delay as [2]

$$\phi = kn\Delta(x,y) - k(\Delta_0 - \Delta(x,y)) \tag{5.47}$$

where the last term in right-hand side of the equation is due to the the propagation in air. Therefore, for an incoming wave $U(x,y)$ we can write

$$\tilde{U}(x,y) = U(x,y)e^{ik\Delta_0}e^{ik(n-1)\Delta(x,y)} \tag{5.48}$$

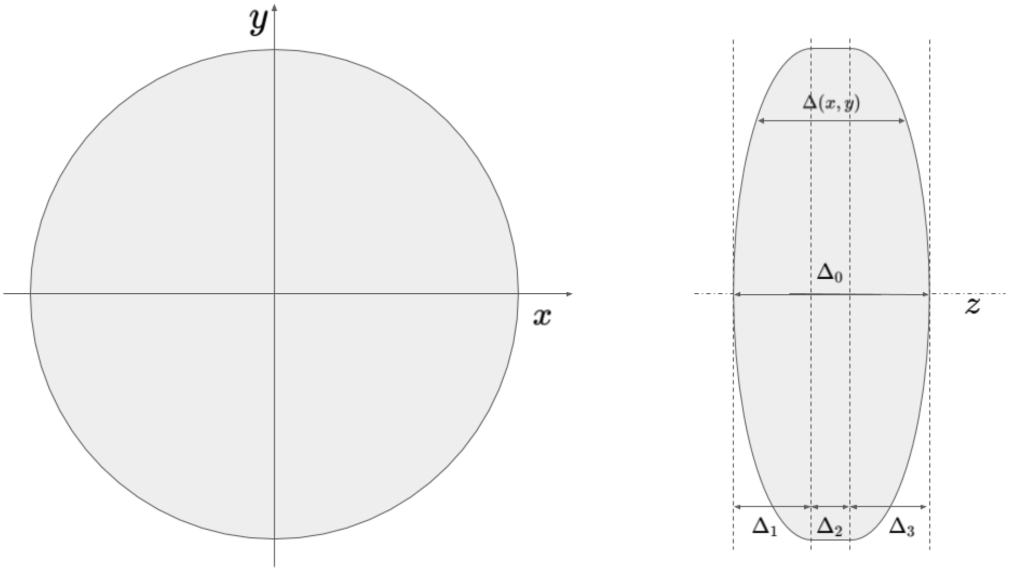


Fig. 5.7: Front (left image) and lateral (right image) view of a lens. Please note the maximum thickness $\Delta_0 = \Delta_1 + \Delta_2 + \Delta_3$ for the whole lens and the lens thickness $\Delta(x, y)$.

From Fig. 5.7, it is hinted that we can rewrite the thickness function $\Delta(x, y)$ as a sum of three components, $d_1(x, y)$, $d_2(x, y)$ and $d_3(x, y)$. Following the geometry in the figure and assuming that the rays of light travel from the left to the right, we can define two numbers $R_l > 0, R_r < 0$ that will be used to take into account the two arcs of circumference that delimit the lateral profile of the lens. Please note that $|R_l|$ and $|R_r|$ are the radii for the leftmost and rightmost arcs, respectively.

First, the thickness function due to the central section of the lens is constant. Therefore,

$$d_2(x, y) = \Delta_2 \tag{5.49}$$

For $d_1(x, y)$, we use R_l and write [2]

$$d_1(x, y) = \Delta_1 - R_l \left(1 - \sqrt{1 - \frac{x^2 + y^2}{R_l^2}}\right) \tag{5.50}$$

To describe $d_3(x, y)$, we simply substitute Δ_2 to Δ_1 and $-R_r$ to R_l (remembering that $(-R_r)^2 = R_r^2$) in Equation (5.50).

Now we can write

$$\Delta(x, y) = d_1(x, y) + \Delta_2 + d_3(x, y) \tag{5.51}$$

The expression for the total phase delay is still too complicated but can be simplified with the introduction of the paraxial approximation, i.e. considering only those rays that lie close to the axis of the lens. In this case, x and y are sufficiently small, and it is possible to expand $d_1(x, y)$ and $d_3(x, y)$ into Taylor series to the first order by using $\sqrt{1+a} \approx 1 + \frac{a}{2} + ...$

Therefore, the thickness function in paraxial approximation becomes [2]

$$\Delta(x, y) = \Delta_0 - \frac{x^2 + y^2}{2} \left(\frac{1}{R_l} - \frac{1}{R_r} \right) \tag{5.52}$$

where we have exploited $\Delta_0 = \Delta_1 + \Delta_2 + \Delta_3$ from the geometry in Fig. 5.7.

Now, we can compute the total phase delay and drop the constant terms as they will be unimportant. For an incoming wave $U(x, y)$, we write

$$\tilde{U}(x, y) = U(x, y) e^{-i \frac{k}{2f}(x^2+y^2)} \tag{5.53}$$

where f is called *focal length* and is defined as

$$\frac{1}{f} = (n-1) \left(\frac{1}{R_l} - \frac{1}{R_r} \right) \tag{5.54}$$

In real lenses, or when the paraxial approximation is not valid, the wave front after the lens will depart from the predicted spherical behaviour. In that case, we will say that the system has aberrations. In this thesis, we will only consider ideal lenses and neglect all aberrations.

Real lenses have a finite aperture, which introduces the *pupil function*, $P(x, y)$. $P(x, y)$ can be interpreted as a mask to the incoming wave. If the point is outside the lens pupil, then it is neglected. In practice, we can write

$$P(x, y) = \begin{cases} 1 & \text{in the aperture} \\ 0 & \text{otherwise} \end{cases} \tag{5.55}$$

The pupil is usually either circular or rectangular. In the former case, we will use the circ function in (5.45) to describe it, while in the latter we will use two one-dimensional rect functions,

$$\text{rect}(t) = \begin{cases} 1 & if\ |t| \le \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.56}$$

To take the pupil into account we simply multiply (5.53) by $P(x, y)$, or

$$\tilde{U}(x, y) = U(x, y) P(x, y) e^{-i \frac{k}{2f}(x^2+y^2)} \tag{5.57}$$

5.5.1 Fourier Transforming Properties of a Lens

Depending on whether the incoming planar wave field is transformed into a converging or diverging spherical wave, lenses are said to be either *converging* or *diverging*.

We will consider a converging lens in a coherent imaging system for this section. Let $U(\xi, \eta)$ be the incoming wave field generated by an input placed at distance d from the lens. The quantity of interest is the wave field at a distance f behind the lens.

Let us now consider the wave at after the lens. By direct substitution in the Fresnel transform and some geometrical considerations [2], we have

$$\begin{aligned} U_f(x, y) = &-\frac{i}{\lambda f} e^{i\frac{k}{2f}\left(1-\frac{d}{f}\right)(x^2+y^2)} \\ &\times \int\int_{\mathbb{R}^2} U(\xi, \eta) P(\xi + \frac{d}{f}x, \eta + \frac{d}{f}y) e^{-i\frac{k}{2f}(x\xi + y\eta)} d\xi d\eta \end{aligned} \tag{5.58}$$

where the quadratic phase terms within the integral cancel out.

The effect introduced by the limited extent of the pupil is known as *vignetting*. If we can neglect the pupil function (i.e. if $P(\xi, \eta) = 1$ everywhere) and place the input at $d = f$, then we see that the output wave field is proportional to the Fourier transform of $U(\xi, \eta)$. Indeed, if $P(\xi, \eta) = 1$, we have

$$U_f(x, y) = -\frac{i}{\lambda f} e^{i\frac{k}{2f}\left(1-\frac{d}{f}\right)(x^2+y^2)} \int\int_{\mathbb{R}^2} U(\xi, \eta) e^{-i\frac{k}{2f}(x\xi + y\eta)} d\xi d\eta \tag{5.59}$$

If $d = f$ the quadratic phase in the exponential in front of the integral vanishes, leaving only the Fourier transform of $U(\xi, \eta)$ multiplied by a constant factor.

Furthermore, if the input is placed at $d = f$, then the phase term outside of the integral disappears, which leaves only the Fourier transform multiplied by the constant term $-\frac{i}{\lambda f}$.

5.5.2 Pupil Transfer Function

We now seek an expression for the pupil function in the Fourier domain as it simplifies computation and offer further insight.

First, we consider the notation in Fig. 5.8.

We can rewrite the imaging system as [2, 22]

$$U_d(x, y) = (U_g * h)(x, y) \tag{5.60}$$

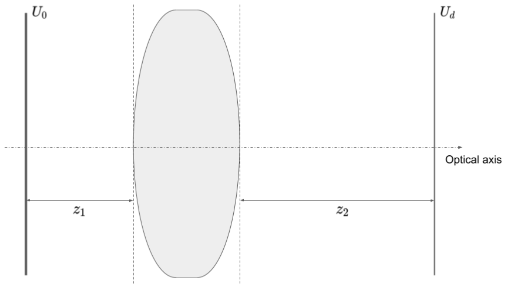


Fig. 5.8: Geometry of the system that we will use. U_0 is the wave field at the object plane, and U_d is the wave field at z_2 behind the lens.

where h is the impulse response and U_g is the ideal field magnified according to geometrical optics,

$$U_g(x, y) = \frac{1}{|M|} U_0 \left(\frac{x}{M}, \frac{y}{M} \right) \tag{5.61}$$

Here, M is the magnification. Practically, the magnified field is blurred by the convolution kernel, which describes the effect of diffraction and the lens. Furthermore, magnification also changes the perceived sampling interval—and thus the Nyquist frequency. We have

$$\Delta x' = \frac{\Delta x}{M} \qquad \Delta y' = \frac{\Delta y}{M} \tag{5.62}$$

where Δx and Δy are the physical sampling intervals along the x and y directions of the detector.

It is known that the condition for the best focus in presence of a lens is given by the so called classical *lens law*, which is

$$\frac{1}{z_1} + \frac{1}{z_2} = \frac{1}{f} \tag{5.63}$$

Under this condition, we can write

$$\begin{aligned} h(x, y) &= \int \int_{\mathbb{R}^2} P(\lambda z_2 u, \lambda z_2 v) e^{-i2\pi(ux+vy)} du dv \\ &= \mathcal{F} \left\{ P(\lambda z_2 u, \lambda z_2 v) \right\} (x, y) \end{aligned} \tag{5.64}$$

with u and v defined according to (5.36).

Let us denote the Fourier transform of h with H. We will say that H is the transfer function of the lens. We can compute it explicitly without any loss of generality. Indeed,

$$\begin{aligned} H(u, v) &= \mathcal{F}\mathcal{F} \left\{ P(\lambda z_2 u, \lambda z_2 v) \right\} (u, v) \\ &= P(-\lambda z_2 u, -\lambda z_2 v) \end{aligned} \tag{5.65}$$

where the last equivalence follows directly from (2.16). Considering the usual shapes of the pupil, which are defined by the circ and rect function, we see that $P(u,v) = P(-u,-v)$.

Then, we can rewrite Equation (5.60) as

$$U_d(x,y) = \mathcal{F}^{-1}\{\mathcal{F}\{U_g\}H\}(x,y) \tag{5.66}$$

The lens pupil is essentially a mask that completely cuts all the frequencies above a certain cut-off frequency. Therefore, we can say that the pupil transfer function $H(u,v)$ is an ideal low-pass filter.

The expression for the cut-off frequency is [22]

$$f_0 = \frac{w}{\lambda z_2} \tag{5.67}$$

where w is the radius of the lens aperture—i.e. the radius of the circumference in the rightmost image in Fig. 5.7.

If the paraxial approximation is valid, we can define the f-number $f/\#$ as

$$f/\# = \frac{z_2}{2w} \tag{5.68}$$

and the numerical aperture NA as

$$\mathrm{NA} \approx \frac{1}{2f/\#} \tag{5.69}$$

Thus, for coherent imaging systems, the cut-off frequency can be expressed without knowing the geometry of the lens aperture,

$$f_0 = \frac{1}{2\lambda f/\#} \approx \frac{\mathrm{NA}}{\lambda} \tag{5.70}$$

For a circular pupil we therefore write [22]

$$H(u,v) = \mathrm{circ}\left(\frac{\sqrt{u^2+v^2}}{f_0}\right) \tag{5.71}$$

The resolution of a system with lenses is limited by diffraction. If it is close to the theoretical limits introduced by the physics behind this phenomenon, we will say that the system is *diffraction limited.*

Practical Considerations for Optical Simulations Based on Lenses

The formulae in (5.66) and (5.67) form the basis for the simulation of an ideal lens, i.e. a lens whose imaging performance can completely be ascribed to the shape of its pupil. Such a lens does not produce any aberration.

We must, however, introduce an inherent limitation to the simulation approach. Let us introduce the sampling intervals Δx and Δy and assume that, for simplicity, $\Delta x = \Delta y$.

Real detectors do not measure $U_d(x, y)$ directly but record its intensity. Now, since $I(x, y) = |U(x, y)|^2$, theoretically the maximum frequency of the intensity can be at most twice the one found in $U(x, y)$ [22]. This can be easily seen by considering a cosine wave $\cos(kx)$ and exploiting the properties of the exponential and Euler equation. We have

$$e^{i2\theta} = \left(e^{i\theta}\right)^2$$

with $\theta = kx$. Now, considering only the real parts, we have

$$\cos(2\theta) = 2\cos^2\theta - 1$$

Therefore, to simulate a lens, we require that

$$2f_0 \leq f_N \tag{5.72}$$

where $f_N = \frac{1}{2\Delta x}$ is the Nyquist frequency as defined by (3.12). This condition ensures that the frequencies in the model will be all contained in a circle of radius f_N in the Fourier domain of the blurred image.

5.5.3 Rayleigh Criterion for Resolution

Under ideal imaging conditions, the intensity of a focused diffracted circular aperture can be computed analytically as [22]

$$I(x, y) = \left(\frac{w^2}{\lambda z}\right)^2 \left(\frac{J_1\left(2\pi \frac{w}{\lambda z}\sqrt{x^2 + y^2}\right)}{\frac{w}{\lambda z}\sqrt{x^2 + y^2}}\right)^2 \tag{5.73}$$

This expression is used to generate the *Airy disk*—or *Airy pattern*—which describes the best focus PSF of an optical system. In the expression, z is the recording distance, w is the aperture radius, λ is the wavelength of the incident light wave and J_1 is the Bessel function of the first kind of order one.

Equation (5.73) is valid if $\frac{w^2}{\lambda z} < 0.1$.

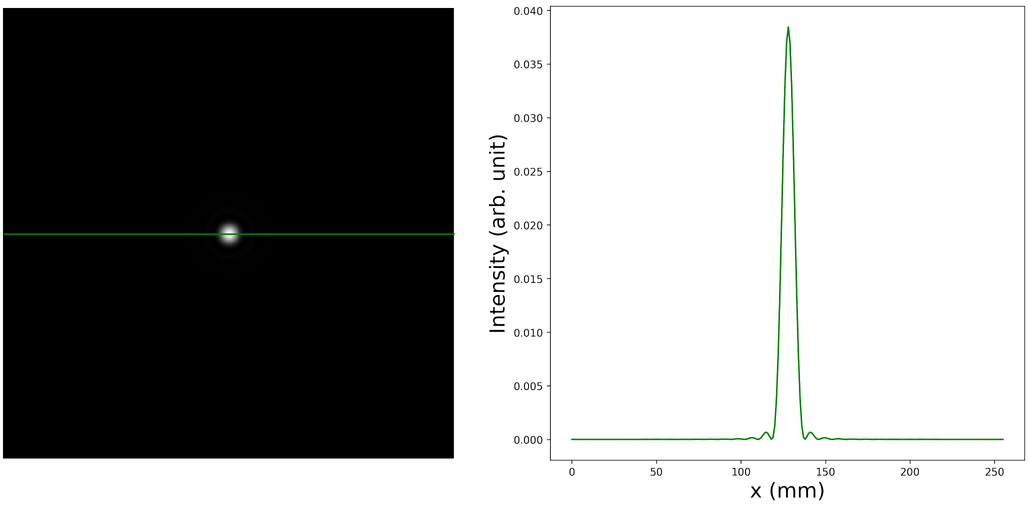


Fig. 5.9: On the left, the Airy pattern with w = 1.375 µm, z = 47.85 µm and λ = 633 nm. On the right, the line profile of the airy pattern passing through $y = 0$ (central row).

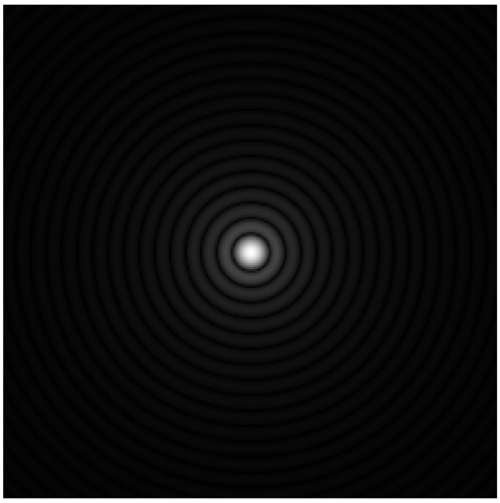


Fig. 5.10: Cube root of the intensity in the left image in Fig. 5.9 to enhance the visualisation of the Airy pattern.

The Airy pattern for w = 1.375 µm, z = 47.85 µm and λ = 633 nm is shown as an example in Fig. 5.9 and Fig. 5.10, where the grey levels have been rescaled by applying the cube root to the airy pattern for better visualisation. The pixel size in the image is $\Delta x = \Delta y =$ 1.375 µm.

The Rayleigh criterion is defined by the first zero of the Bessel function $J_1(a)$, which is observed when the $a = 1.22\pi$. To simplify calculations, we can set $y = 0$ and find the half-diameter δ of the disk

$$\delta = 0.61\frac{\lambda z}{w} = 0.61\frac{\lambda}{\mathrm{NA}} \tag{5.74}$$

δ is also the distance for which two diffracted point disturbances are barely resolved.

5.6 Imaging With Lenses and Defocus

Sometimes, the imaging system in Fig. 5.8 is not perfectly focused. In those cases, we can introduce a third variable d, which is the lens-detector distance, and isolate the effect of the lens and defocus into a kernel,

$$h(x, y; d) = -\frac{1}{\lambda^2 z_1 d}\int\int_{\mathbb{R}^2} P(\xi, \eta) e^{ik\left(\frac{1}{d}-\frac{1}{z_2}\right)(\xi^2+\eta^2)} e^{-i\frac{k}{d}(\xi x+\eta y)} d\xi d\eta \tag{5.75}$$

where we have used the same notation introduced in Fig. 5.8.

Equation (5.75) can be interpreted as the PSF of a defocused imaging system without aberrations. If the pupil is circularly symmetric, the Fourier transform can be rewritten as

$$\mathcal{F}\{g(r, \theta)\} = 2\pi\int_0^{+\infty} r g_R(r) J_0(2\pi r\rho) dr$$

for any $g : \mathbb{R}^2 \to \mathbb{C}, g_R : \mathbb{R} \to \mathbb{C}$ such that g is circularly symmetric, namely $g(r, \theta) = g_R(r)$, where (r, θ) are polar coordinates [2].

Chapter 6

Holography

Holography is an application of coherent imaging that requires the measurement of the intensity of the interference pattern of a background reference wave and the wave diffracted by the object. Reconstruction of the hologram from the photographic film can be achieved by illuminating the film with the same reference wave used during acquisition.

Mathematically, we can represent the intensity on the detector as

$$\begin{aligned} I(x,y) &= |U_R(x,y) + U_O(x,y)|^2 \\ &= |U_R(x,y)|^2 + |U_O(x,y)|^2 + \overline{U_R}(x,y)U_O(x,y) \\ &\quad + U_R(x,y)\overline{U_O}(x,y) \end{aligned} \tag{6.1}$$

where $U_R(x,y)$ and $U_O(x,y)$ are the complex-valued reference and object wave.

Borrowing the result for the interference of two waves in Equation (5.13), we can rewrite the intensity in (6.3) as

$$I(x,y) = A(x,y) + B(x,y)\cos\left(\Delta\phi(x,y)\right) \tag{6.2}$$

where $A(x,y) = |U_R(x,y)|^2 + |U_O(x,y)|^2$, $B(x,y) = 2|U_R(x,y)||U_O(x,y)|$, and $\Delta\phi(x,y)$ is the phase difference between the two waves. Please note that the phase is encoded into the intensity.

Illuminating the film with the same illumination wave—or reference wave—gives

$$\begin{aligned} \psi(x,y) &= U_R(x,y)I(x,y) \\ &= U_R(x,y)|U_R(x,y)|^2 + U_R(x,y)|U_O(x,y)|^2 \\ &\quad + U_R(x,y)\overline{U_R}(x,y)U_O(x,y) + U_R(x,y)U_R(x,y)\overline{U_O}(x,y) \end{aligned} \tag{6.3}$$

The first two terms, $U_R(x,y)|U_R(x,y)|^2 + U_R(x,y)|U_O(x,y)|^2$, constitute the DC component or zero-order term of the hologram. The twin image—or virtual image—is represented by $|U_R(x,y)|^2 U_O(x,y)$ and is reconstructed in Equation (6.3). The last term, $U_R(x,y)U_R(x,y)\overline{U_O}(x,y)$, represents the *distorted* real image. To reconstruct an undistorted real image, we simply have to illuminate the plate with $\overline{U_R}(x,y)$.

6.1 Hologram Recording and Reconstruction

Traditionally, holograms were recorded using the *in-line hologram* acquisition method. In this experimental approach, the reference wave $U_R(x,y)$ travels in the same direction of the optical axis, which is perpendicular to the photographic film (or, later, a detector). The object is placed between the coherent light source and photographic film and diffracts the incoming reference wave. This diffracted wave interferes with $U_R(x,y)$ on the recording medium.

Reconstruction via illumination with the reference wave produces a twin image at a distance d behind the photographic plate and a real image at a distance d in front of the plate.

Nowadays, the photographic film has been replaced by a detector (e.g. CMOS, CCD) and the holograms are reconstructed numerically [25–27]. However, in in-line holography, we have a superposition of the DC term and the twin and real images, limiting resolution and overall image quality.

Another approach to holography is *off-axis* holography [28]. In this technique, the reference and object waves are separate and tilted with respect to each other. In particular, the reference beam is tilted with respect to the detector by a certain angle θ. The choice of the tilt angle is limited by the Nyquist frequency of the recording medium to avoid aliasing [21].

It has been shown that off-axis holography produces an almost homogeneous pattern onto the photographic plate and is less sensitive to film imperfections or damage [28], while these defects cause loss of details in other holographic techniques. This property is still present with digital images acquired in off-axis configuration, as most of the image can be deleted without corrupting the reconstructed hologram significantly. It may also be said that the information about the object is encoded in each and every point of the hologram.

In this configuration, the zero-order term and the twin and real images are spatially separated. However, if digital reconstruction is chosen, only one of the two images will be focused after reconstruction, while the other will be out-of-focus. Furthermore, in off-axis digital holography it is also possible to suppress the zero-order term [29, 30].

There is another approach to digital holography, known as Phase Shifting (PS) Digital Holography (DH) [31,32], in which either the reference or the object wave are phase stepped with the aid of a piezoelectric transducer (PZT) and the phase-shifted digital holograms are linearly combined to obtain the complex wave field. This method automatically removes the zero-order term and the twin image. The wave field can be refocused with any numerical diffraction algorithm such as (5.34), (5.39) and (5.46) provided the approximations hold.

PS-DH can be obtained with as few as three holograms. Due to the simplicity of the method, we will now introduce the 4-step PS-DH with a phase step of $\frac{\pi}{2}$. The dataset is composed of four holograms, and, assuming that the PZT shifts the reference beam, we have

$$I_n(x,y) = |U_R(x,y)e^{in\frac{\pi}{2}} + U_O(x,y)|^2 \qquad n = 0,1,2,3 \tag{6.4}$$

where we have used the same notation introduced in (6.1).

We can shift a wave with

$$E(x,y) = I_0(x,y) - I_2(x,y) + i\,(I_1(x,y) - I_3(x,y)) \tag{6.5}$$

and obtain the complex object wave with

$$U_O(x,y) = \frac{E(x,y)}{4\overline{U_R}(x,y)} \tag{6.6}$$

Please note that $4\overline{U_R}(x,y)$ can be obtained by removing the object and measuring the phase-stepped object-free interference patterns on the detector with PS-DH. Equation (6.5) applied to these four reference holograms should give $4U_R(x,y)$.

If we only have the first three I_n, we can still obtain the object wave with

$$U_O(x,y) = \frac{1-i}{4\overline{U_R}(x,y)}\,(I_0(x,y) - I_2(x,y) + i\,(I_1(x,y) - I_2(x,y))) \tag{6.7}$$

Please note that, in theory, acquisition of a digital hologram at the best focus condition is not a requirement because it can always be refocused numerically.

6.2 Phase Unwrapping

It has been shown that Quantitative Phase Imaging (QPI) is feasible with digital holography [33, 34]. However, we have to remember that the phase is not measured directly but has to be extracted from the reconstructed complex wave field $U(x,y)$ as

$$\phi_w(x,y) = \arctan\left(\frac{\Im\{U(x,y)\}}{\Re\{U(x,y)\}}\right) \tag{6.8}$$

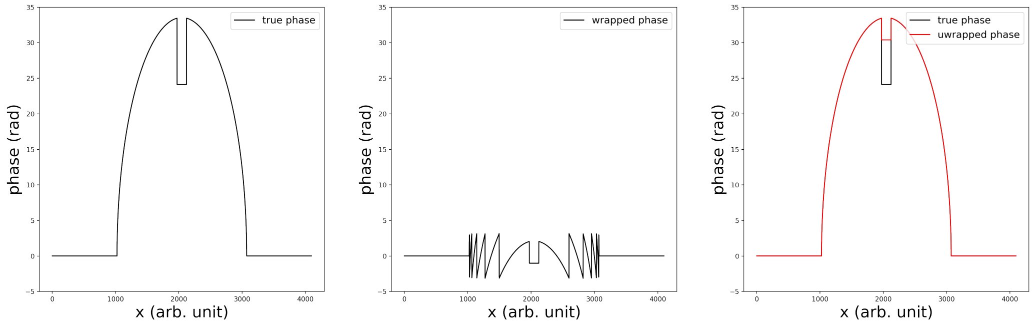
where $\Re$ and $\Im$ are used to extract the real and imaginary part of $U(x,y)$, respectively. Alternately, $\phi_w(x,y)$ can be obtained from the two-argument arctangent $\arctan 2$, which maps $\phi_w(x,y)$ into $[-\pi,\pi]$ instead of $[-\frac{\pi}{2},\frac{\pi}{2}]$. From now on, we will always assume that the extracted phase $\phi_w(x,y)$ will be mapped to the interval $[-\pi,\pi]$.

Please note that $\phi_w(x,y)$ is usually not the real phase of the wave field but only its *wrapped* phase. In fact, it is not possible to extract the true phase with an inverse trigonometric function because the complex exponential has from 2π ambiguity,

$$e^{i\phi} = e^{i(\phi+2n\pi)} \quad \forall n,\phi \mid \; n \in \mathbb{Z},\; \phi \in \mathbb{R} \tag{6.9}$$

From Equation (6.9), we notice that if there is a constant $\phi_0 \in \mathbb{R}$ for which the true phase $\phi(x,y)$ of the wave field is completely defined in an interval $[\phi_0, \phi_0 + 2\pi[$, then we can say that the wrapped phase is equal to the true phase up to a constant.

Unfortunately, it is not uncommon to have a true phase that spans tens of π. In this case, the true phase must be retrieved from the wrapped phase, and we say that the phase has to be *unwrapped*. This problem and the algorithms that attempts to provide a solution to it are called *phase unwrapping*. Several solutions are available [35–38].



(a) True phase.

(b) Wrapped phase.

(c) Comparison of the true and the unwrapped phases.

Fig. 6.1: Please note that phase unwrapping fails to retrieve the true phase because the information of the true phase discontinuity has already been lost during wrapping, as it can be clearly seen in Fig. 6.1b.

Phase unwrapping has inherent limitations. In particular, if the true phase, $\phi(x,y)$, of two adjacent points differs by at least 2π, then in general it is not possible to estimate it via phase unwrapping. This is true regardless of whether the phase distribution is continuous or presents discontinuities. To illustrate this problem, we offer the one-dimensional phase distribution in Fig. 6.1a. Please note that the information about the true value of the jump is lost during wrapping as it is greater than 2π (Fig. 6.1b) and that the unwrapping algorithm therefore fails to recover the true phase (Fig. 6.1c).

6.3 Simulation of an Hologram

The simulation of an hologram follows straightforwardly from the theory on diffraction [39, 40].

In fact, if the wave at the object plane—or any other plane—is known, the diffracted wave at any recording distance can be computed numerically with, for example, the Fresnel transform in (5.34) or the angular spectrum method in (5.46) by computing (5.44) with the imaging system parameters.

Furthermore, the effect of an ideal lens can be simulated by clipping the Fourier domain according to Equation (5.66) with the lens transfer function (5.65), which is completely determined by the lens pupil. The problem of magnification is solved by Equation (5.61).

The true problem is now the simulation of the object wave, which can, in general, be written as

$$U_{Obj}(x, y) = U_0(x, y)e^{-a(x,y)}e^{i\phi(x,y)} \tag{6.10}$$

where U_0 is the incident beam at the object plane, which is also used as a reference wave for in-line holography. If the hologram is acquired with PS-DH and the incoming reference beam is split into two beams of half the initial intensity each, then the reference and incident beams will be identical up to the phase.

The function $\phi\colon \mathbb{R}^2 \to \mathbb{R}$ is the phase accumulated by the object wave, while $a\colon \mathbb{R} \to [0, 1]$ represents the attenuation of the incident wave. Please note that if $a(x, y) = 0$ and $\phi(x, y) \neq 0$, then we have a *pure phase object*. Analogously, if $\phi(x, y) = 0$ and $a(x, y) \neq 0$, then only the attenuation is of interest, and we have a *pure amplitude object*.

Please note that when $a(x, y) = 0$ and $\phi(x, y) = 0$, we either have the undisturbed incident wave or no object at all. This means that we can rewrite the object wave as

$$U_{Obj}(x, y) = U_0(x, y) + U_1(x, y) \tag{6.11}$$

where $U_1(x, y)$ contains all and only the effects ascribed to the object and vanishes where the incident wave is left undisturbed or there is no object.

$U_O(x, y)$ can be calculated via analytical formulae or models, if available, or via simulation.

To get the hologram, we have to take the intensity of the interference pattern according to (6.1). Please note that we simply take the intensity of the numerically diffracted $U_{Obj}(x, y)$ in in-line holography. In off-axis holography and in PS-DH, we have to add the reference wave to $U_{Obj}(x, y)$ before computing the intensity.

Please note that if the chosen acquisition strategy is the PS-DH, it is also possible to retrieve the object wave by employing the correct phase shifting formula. Therefore, if $U_{Obj}(x, y)$ is the quantity of interest, we can skip the calculation of the intensity.

Chapter 7

Optical Diffraction Tomography

Optical Diffraction Tomography (ODT) is a phase-contrast tomographic technique that aims to reconstruct the three-dimensional RI distribution of a sample. The projections are acquired through digital holography and processed to produce the sinograms of the phase distribution of the object. It has found applications in cell imaging [41, 42] and the investigation of large-scale (millimetre-size) samples [43–45].

CT reconstruction can be performed with FBP [44] and diffraction tomography [43, 46].

7.1 Apparatus

The highly configurable Mach-Zehnder interferometer, shown in Fig. 7.1, is an excellent choice for an application like ODT.

A monochromatic light beam is produced by the laser (L) and is expanded by the beam expander (BE) before being split into two waves: one for the reference arm and one for the object arm.

On the reference arm, the beam is guided via mirrors (M) and phase shifted by a piezoelectric transducer (PZT) by steps of $\frac{\pi}{2}$ before being magnified.

On the sample arm, the incoming beam is diffracted by the sample (S)—which is placed on the rotating stage (RS) to allow illumination at different angles for tomography. The diffracted wave is magnified by the magnifying object (MO) and interferes with the reference wave on the second beam splitter (BS).

The intensity of the interference pattern is recorded by the detector (D), which is placed after the tube lens (TL).

The acquisition of three or more holograms phase stepped by $\frac{\pi}{2}$ makes

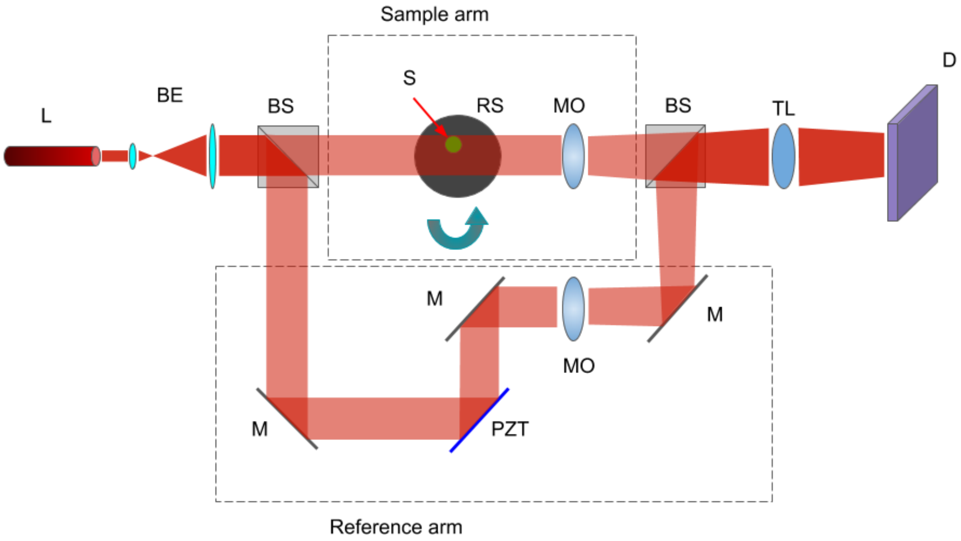


Fig. 7.1: A Mach-Zehnder interferometer for ODT via PS-DH. In figure: L = laser; BE = beam expander, BS = beam splitter; S = sample; RS = rotating stage; MO = magnifying object; M = mirror; PZT = piezoelectric transducer; TL = tube lens; D = detector.

the reconstruction of the complex wave field possible via linear combination.

The apparatus in Fig. 7.1 is a $4f$ system due to the MO and TL between the sample and the detector. Remembering the effect of the lens on an incoming wave (5.58) and neglecting the pupil as in (5.59), we notice that if we place the sample at the focal length of the MO, the detector at the focal length of the TL and the MO and TL at twice the focal length from each other, we effectively measure the intensity of $\mathcal{F}\mathcal{F}U(x, y) = U(-x, -y)$, which is the inverted image.

7.2 Sample Placement and Rotation

The sample does not need to be placed at the best focus plane as it can be refocused numerically.

In particular, off-centre placement of the sample has shown to improve the Signal-to-Noise Ratio (SNR) of the reconstructed slice [47]. Please note that this changes the recording distance, which, however, does not need to be measured because it can be computed via focus scanning.

Numerical focus scanning consist in propagating a complex wave by a set of distances $\{d_n\}$ and evaluating the focus of the whole input or a Region of Interest (ROI) at each recording distance with a metric. The recording distance that offers the best focus according to the metric is kept. It has been shown that the variance of the absolute value of the wave field is a good metric for ODT [44].

Focus scanning is inherently computationally taxing, especially if it has to be performed on all the projections. Fortunately, it is possible to find a general expression for the reconstruction distance by exploiting the geometry in Fig. 7.1 and introducing the coordinate system as in Fig. 7.2, where we assume that z is the optical axis.

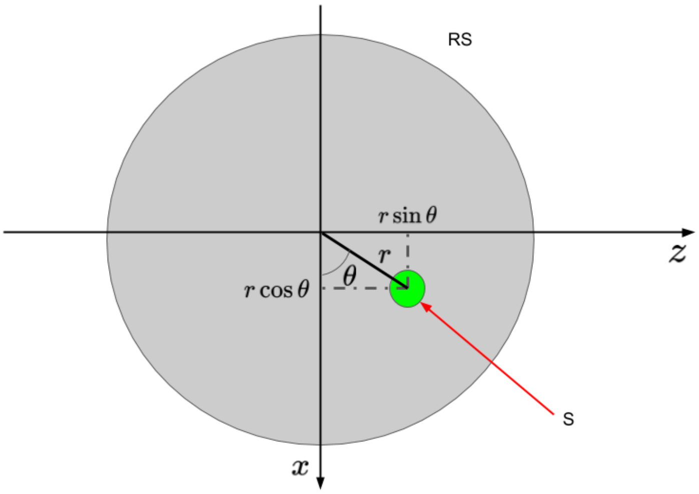


Fig. 7.2: Coordinate system of the rotating sample (S). The origin is in the centre of rotation, which is the centre of the rotating stage (RS).

In general, we expect that the plane of best focus will be at a distance d_0 from the centre of rotation. If the plane in which lies the centre of rotation (i.e. $z = 0$) and the plane of best focus coincide, we have $d_0 = 0$.

Now, physically, the sample rotates about the centre of the RS, which means that it moves either closer or further away from focus. If $d_0 = 0$, there will be two points in which the object will be at focus. If $d_0 \neq 0$, the object might never be fully focused depending on r and d_0.

From Fig. 7.2, it is clear that the recording distance d depends on the projection of r (the distance of the sample from the centre of rotation) onto the optical axis z. The most general expression for the recording distance is

$$d(\theta; d_0, r, b, \theta_0) = r \sin(b\theta + \theta_0) + d_0 \tag{7.1}$$

where b is the scale factor on the angle θ and θ_0 is the angle at the initial position of the sample. This angle is 0 only when the object lies on the x-axis at the beginning of the acquisition.

The strategy to speed up the focus scanning is [44]:

Step 1: extract a subset of complex wave fields from the dataset of complex wave fields recorded at all angles. Please note that the complex-valued images in this subset must be distributed over the interval $[0, 2\pi]$ more or less uniformly. For example, we may extract one image every $\frac{2\pi}{10}$ rad;

Step 2: choose two numbers $d_b, d_e \in \mathbb{R} \mid d_b < d_e$ such that the true recording distance should be in the the range $[d_b, d_e]$. Select a number of steps N and sample the interval $[db, d_e]$ into N steps;

Step 3: for every wave field $W(x, y)$ and for every sampled distance $\tilde{d}$, propagate $W(x, y)$ by $\tilde{d}$ with a numerical diffraction routine and evaluate the focus with the focus metric. Store the results;

Step 4: for every wave field, find the best focus by studying the collected results. In practice, there should be a criterion—based on the result of the metric—that determines whether a recording distance refocuses more than another one. Therefore, for each wave field, any distance $\tilde{d}$ is compared against all the others. The best one according to the criterion is kept;

Step 5: fit the distances estimated from the previous step with (7.1).

For ODT images, the expected recording distance can be found as the distance for which the variance of the absolute value of the wave field (or of a ROI of the wave field) is minimum [44].

7.3 Phase-Refractive Index Relation

In ODT, the phase is retrieved from the complex wave field of the object and unwrapped. Then, phase sinograms are formed from the processed projections and reconstructed via a reconstruction algorithm of choice. This implies that the reconstructed slices measures the phase distribution and not the RI. Fortunately, it is possible to retrieve the variation of RI via normalisation. In fact, if $\phi(x, y)$ is the value of a pixel in a reconstructed slice, its RI is [47]

$$\Delta n(x, y) = \frac{1}{k\Delta x}\phi(x, y) = \frac{\lambda}{2\pi\Delta x}\phi(x, y) \tag{7.2}$$

were k is the wave number, λ is the wavelength of the incident plane wave and Δx is the sampling interval.

The true refractive index can be retrieved via

$$n(x,y) = n_0 + \Delta n(x,y) \tag{7.3}$$

where n_0 is the RI of the surrounding medium.

If the specimen spans several slices, then phase and RI in Equations (7.2) and (7.3) will also depend on z. The expressions remain otherwise unchanged.

Previously, the limitations of phase unwrapping and QPI have been shown for a two dimensional phase distribution. Similar considerations are also true for ODT, where the processed projections measure the accumulated phase within the sample. This, theoretically, limits the kind of samples that can be imaged via ODT either to those with very small thickness or to those very small Δn with respect to the surrounding medium. Fortunately, it is possible to lower the RI of a specimen via *optical clearing* and to place it in a similarly refracting medium during acquisition. This last technique is called *refractive index matching.*

Part II

Methods and Results

Chapter 8

Optical Diffraction Tomography Processing Pipeline

The reconstruction pipeline for an ODT dataset acquired with four-step phase-shifting digital holography of an off-centre specimen can be summarised into

1. focus scanning of a subset of the dataset and fit to obtain the reconstruction distance at all angles;

2. numerical propagation to focus holograms;

3. phase extraction and unwrapping;

4. sinogram generation via reslicing of the image stack;

5. parallel beam FBP of the sinograms.

Due to the large amount of required memory for a naive implementation, a pipeline that used batch processing was implemented in Python and C++ to fix this limitation. To offer a gauge to estimate the required memory, we would like to remark that the wave field associated to a simple 12 MPx image (i.e. an image acquired by a detector with 4096×3072 pixels) requires 192 megabytes of RAM just to be loaded: indeed, each pixel is a complex number that requires 16 bytes to be represented with double floating point precision. If the precision is lowered, it still requires 96 megabytes.

Almost everything in the list was implemented in Python using optimised libraries like *numpy, scipy* and *scikit-image*. However, the FBP routine was re-implemented in C++ to improve its performance. The sinogram generation was also written in C++.

The pipeline was parallelised using *joblib* for Python and *openMP* for C++.

Numerical propagation was coded using the angular spectrum method (Equations (5.43) and (5.46)), which works even for short recording distances unlike the Fresnel transform (Equation (5.34)). The phase unwrapping was performed via Ghiglia-Romero's unweighed least squares algorithm [36]. Focus scanning is performed by numerically propagating the wavefront, measuring the focus with a metric and comparing the results [48]. In the case of ODT (i.e. pure phase-objects) correct reconstruction distance can be determined as the minimum of the the variance of the modulus of the wave [44,47]. Focus scanning can be performed on a subset of the original dataset and fit the results with a sinusoidal function to improve speed [44].

Type	Description
CPU	Intel core i5-8250U @ 1.60 GHz, turbo boost 3.40 GHz, 4C/8T
OS	Windows 10
RAM	24 (8+16) GB DDR4 2400 MHz
Storage	128 GB SSD (OS) + 1 TB HDD (5400 rpm)

Table 8.1: Laptop data sheet.

The pipeline was tested on a Dell Inspiron 7570 laptop whose specifications are reported in Tab. 8.1. Due to the lack of proper ventilation of the system, the speed was measured on a subset of the images to avoid thermal throttling, which would influence the quantitative comparison of the results. The C++ source code was fed to the Visual Studio 16 (2019) generator via CMake 3.17 and compiled with the Microsoft Visual Studio compiler, which restricted *openMP* support to the version 2.0.

Part	Old pipeline (s)	New pipeline (s)
Refocusing and phase unwrapping	8.2 ± 0.3	7.80 ± 0.04
FBP	40.0 ± 1.2	34.0 ± 0.4

Table 8.2: Results, per image, with 8 threads in all tests. Projection height=3072 pixels, width=4096 pixels. Number of projections=1440.

Since the fit is not always required and can be performed once and reloaded later, only the numerical propagation, together with phase unwrapping, and the FBP were timed. The performance of the new pipeline was compared to the old Matlab script and the average result per image is shown in Tab. 8.2. To obtain the results in Tab. 8.2, a subset of 40 images of size 3072x4096 pixels was used for the first part, and it was executed a few times so as to generate an average result. FBP was tested using a Ram-Lak (ramp) filter and 1440 projections generated by a Shepp-Logan phantom of size a 4096x4096 pixels.

In particular, speed-up was significant in the routines written from scratch in C++. On the other hand, the use of some built-in functions prevented *joblib* from using the *threading* parallel back-end during phase unwrapping, which limited the performance improvement.

Please note, however, that the new pipeline introduces several differences with respect to the old script:

- parallelisation of the first three steps is done on the batch and not on the single image. This should improve performance, but requires a larger memory usage per batch. However, the batch size is user-defined. Please note that the parallelisation on the image is limited by the number of pixels, while the parallelisation on the batch is limited on the batch size, and speed-up is determined by the load. FBP is parallelised n the sinogram instead;

- batch processing is done with frequent input-output operations to the hard-disk drive (HDD), which is slower than keeping everything in RAM and introduces the explicit sinogram generation step, which is simply a selection of a matrix from the 3D array in the old script. This choice, however, stores all the steps into the HDD. Therefore, the reconstruction could be restarted from any key step (fit parameters, unwrapped phase, sinogram). To achieve the same flexibility, data must be written to disk even in the old script;

- since everything is done via batch processing, there is no need to downscale the unwrapped phase via interpolation in order to fit all the projections into RAM or to cut regions of interest (ROIs) from the projections by manually changing the script;

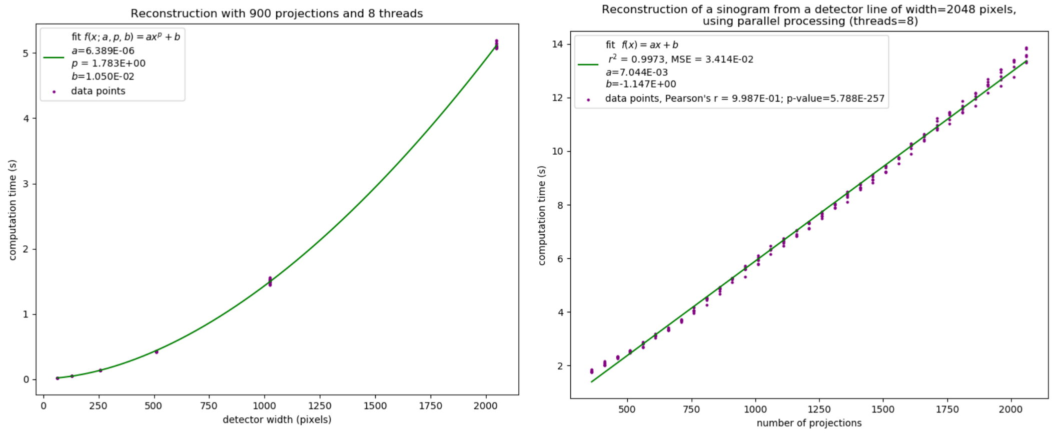
- the number of projections is not limited by the RAM size. Since we can avoid storing all the projections in RAM, the memory usage is proportional to the batch size.

  For example, for 1440 phase projections of shape 4092 $\times$ 3072 pixels, we would need 67.5 gigabytes if each pixel is represented by a single precision (32-bit) floating point number. Please keep in mind that, in this case, 1440 projections are less than recommended and that this is the minimum required memory to store them in RAM. Any operation on the data will likely require additional memory. In this case, we did not consider 16-byte complex numbers because we can process one image at a time. On the other hand, if we work in batches, we first process complex-valued images simultaneously and then extract the phase into 32-bit float arrays. In the first part, for a batch of a 40 images we require 7.5 gigabytes for loading, while in the latter only 1.85 gigabytes are needed. Any processing likely increases memory usage, although it still remains much lower than the naive implementation;

- the variables are not hard-coded, and the pipeline can be run from command line with custom parameters.

The parallel beam FBP algorithm strongly depends on the number of pixels on the detector width and the number of acquired projections.

8.1 Complexity of Filtered Back Projection



(a) Computational time with respect to the number of pixel on a detector row.

(b) Computational time for a detector row of 2048 pixels with respect to the number of projections.

Fig. 8.1: Running time of FBP.

Let N be the number of pixels on a detector row and S be the number of acquired projections. The slice reconstructed via FBP is a square image of side N, and therefore it has N^2 pixels. Since—for each pixel in the reconstruction—we get the position on the detector line and combine all the projections, we expect that the complexity of the FBP is $O(N^2S)$. Please note, however, that the algorithm has additional checks and does not reconstruct pixels outside of the circle of reconstruction—i.e. pixels whose distance from the centre of the slice is bigger than N, which means that only $\pi\left(\frac{N}{2}\right)^2$ pixels will be computed.

To get an estimate of the performance of the algorithm, we set the number of threads to 8 and do two experiments. In the first experiment, we set the number of projections to 900 and set $N = 2^{6+j}$, $j = 0, 1, ..., 5$. We then generate a Shepp-Logan phantom in a square image of side N.

For the second experiment, we set N to 2048 and generate the Shepp-Logan phantom. We vary S from 360 to 2060 in steps of 50, and let the phantom rotate over 2π.

For both experiments, we measure the running time and plot it against the variable of interest in Fig. 8.1a and Fig. 8.1b respectively.

From the results in Fig. 8.1, we see a linear proportionality between the number of projections and the running time. However, the dependence on the number of pixels is slightly off.

8.2 Noise Introduced by FBP

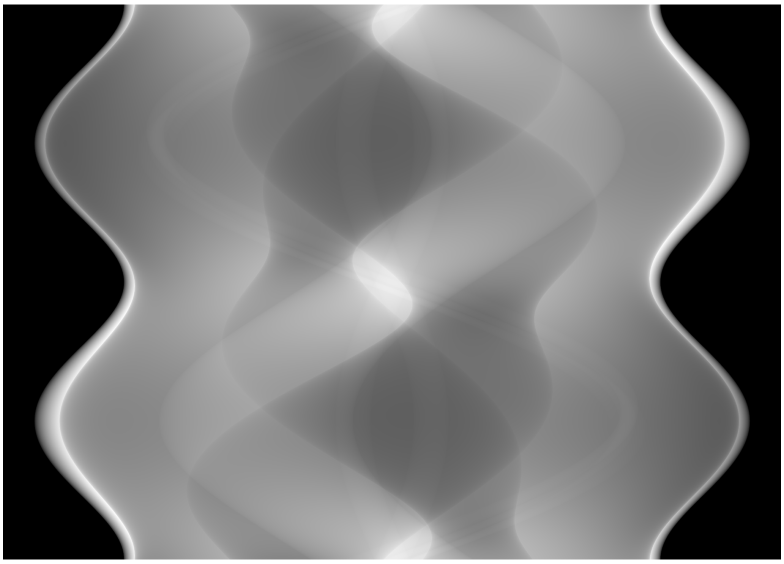


Fig. 8.2: A sinogram of the Shepp-Logan phantom generated from 1460 projections onto a detector row of 2048 pixels.

FBP is a numerical algorithm that uses FFT, linear interpolation, sums and multiplication of floating point numbers. Therefore, it introduces inaccuracies. We use a 2048x2048 Shepp-Logan phantom and vary the number of projections from 360 to 2060 in steps of 50 to test the algorithm. The acquisition of the phantom is performed over 2π, although acquisition in the interval $[0,\, \pi]$ is sufficient in parallel beam FBP.

The Shepp-Logan phantom was chosen for this simulations because it is possible to write the analytical expression for the sinogram, and thus we avoid its generation via discrete Radon transform.

For an ellipse of major axis A, minor axis B, coordinates of the centre c_x and c_y and measurable quantity ρ, we let α the tilt angle between the major axis and the x-axis and set $s = \sqrt{c_x^2 + c_y^2}$ and $\gamma = \tan^{-1}\left(\frac{c_y}{c_x}\right)$.

Let t be the position on the detector and let $t = 0$ be its centre. If θ is the angle of rotation, then let $\phi = \theta - \alpha$ and $\tilde{t} = t - s\cos(\gamma - \theta)$, then the projection of an ellipse for the coordinates (t, θ) in the sinogram is given by [15]

$$P_\theta(t) = P_\phi(\tilde{t}) = \begin{cases} \frac{2\rho AB}{a^2(\phi)}\sqrt{a^2(\phi) - \tilde{t}^2} & if \quad |\tilde{t}| \leq a(\phi) \\ 0 & otherwise \end{cases} \tag{8.1}$$

where $a^2(\phi) = A^2\cos^2\phi + B^2\sin^2\phi$.

An example of sinogram for the Shepp-Logan phantom is shown in Fig. 8.2.

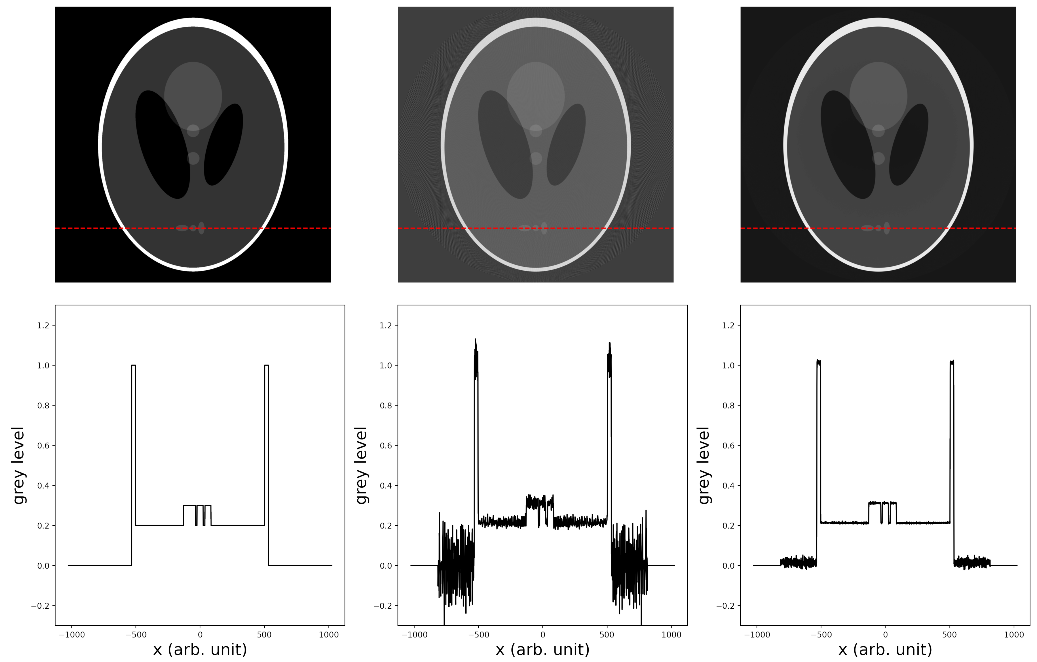
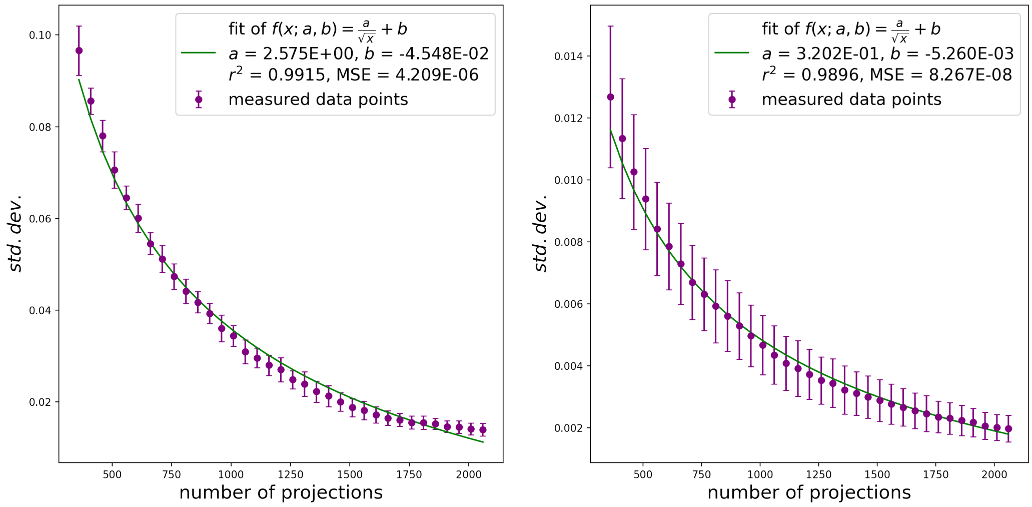


Fig. 8.3: Top row: comparison of the ideal image (left), the image reconstructed from 360 projection (centre) and from 2060 projections (right). Bottom row: the line profile extracted from the dashed line.

Fig. 8.3 shows a comparison of the quality of the reconstruction with respect to noise when the number of projections increases. The leftmost column is the ground truth. Please note the stark improvement between $S = 360$ and $S = 2060$. Since the CT aims to reconstruct the true values that describe the object, an improvement in noise can easily be translated in an improvement in SNR (Signal-to-Noise Ratio).



(a) Background noise when varying the number of projections.

(b) Noise in tissue when varying the number of projections.

Fig. 8.4: Noise behaviour dependence on the number of projections for (a) the background and (b) tissue.

To offer a more quantitative description of the behaviour of noise when the number of projections is increased, we can take the standard deviation of several ROIs (Regions of Interest) within the same material and plot the results against the number of projections. For our test, we took 10 ROIs of about 100×100 pixels from the background and the dark grey tissue (true grey level = 0.2 in Fig. 8.3). The results for the background and the tissue are shown in Fig. 8.4a and Fig. 8.4b respectively. The measured data points were fitted with $f(x) = \frac{a}{\sqrt{x}} + b$, and the goodness of fit is evaluated via r^2 and mean squared error. A similar behaviour was observed with real ODT data [44].

Please note that here r^2 can be used because the model is linear in the parameters a and b. Furthermore, while still present with $S = 2060$ projections, noise in tissue is decreased roughly by a factor of 5 with respect to $S = 360$. That means that if the estimates of the grey level of the tissue are roughly the same, the SNR for $S = 2060$ is about 5 times the one obtained with only 360 projections.

Chapter 9

Simulated PSF

Real, extended objects seldom lie on a single plane. Therefore, only a part of a sample can be focused at a time. This is a known problem in optics applications. For example, simulations based on the beam shape have been used in THz CT [49] and optical projection tomography (OPT) [50], and simulation of points and deconvolution has been also applied in OPT [51] to model and correct space-variant blur.

In coherent imaging the blur depends on the distance from the plane of best focus. Therefore, we expect non-constant blur throughout the image of the sample. While this problem is hard to tackle in general, in the case of a point-like object, we can greatly simplify calculations and models.

This chapter is split into two section: the first deals with the model itself and the second provides the results obtained from the simulation.

9.1 The Model

Let us suppose that we have a plane wave $U_L(x, y)$ that has to be propagated in free space by a distance $z = z_0 + z_1$. Then the propagation in free space can be always written via a convolution,

$$\begin{aligned} U_L(x, y; z_0 + z_1) &= U_L(x, y; 0) * h(x, y; z_0 + z_1) \\ &= U_L(x, y; 0) * h(x, y; z_0) * h(x, y; z_1) \end{aligned} \tag{9.1}$$

It can be shown that the second equivalence holds either by using the angular spectrum method or the operator notation for the Fresnel transform in [2]. Therefore, the propagation in free space by z can always be split in n convolutions with $n \in \mathbb{N}$ arbitrarily large.

In the case at hand, it is most convenient to define a quantity $U_0(x, y)$ that represents the incident plane wave at the focused plane and work directly with it. In this case, working with the incident plane wave at the origin is not necessary and can be avoided throughout calculations as (9.1) is valid.

To simplify calculations, we will assume that the incident plane wave will move along the optical axis—i.e. it is constant if the position on the optical axis is fixed or,

$$U_0(x, y; z) = U_0(z) = u_0 e^{i(kz+\phi_0)} \tag{9.2}$$

with the usual notation. For simplicity we can either drop the constant ϕ_0 or incorporate it in u_0 as it does not provide any insight on the model.

A point-like object is small enough to lie on a single plane, so we can always refocus it with numerical diffraction. Therefore, we start by considering an object wave at the object plane as

$$U_{Obj}(x, y; z) = U_0(x, y; z) + U_1(x, y) \tag{9.3}$$

where $U_1(x, y)$ is a point disturbance at z. To consider any diffraction effect, we simply convolve U_{Obj} with a convolution kernel by a distance z_d and write

$$\begin{aligned} U_{Obj}(x, y; z + z_d) &= U_{Obj}(x, y; z) * h(x, y; z_d) \\ &= U_0(x, y; z) * h(x, y; z_d) + U_1(x, y) * h(x, y; z_d) \end{aligned} \tag{9.4}$$

due to the linearity of the convolution.

If we now consider that $U_0(x, y; z)$ is constant if z is fixed, we see that the first term of Equation (9.4) is constant and describes the background wave and that the object wave is computed as the interference of the diffracted disturbance with the uniform background. Furthermore, if we put an infinitely large blocking plate at the disturbance plane and we left a hole at the coordinates of the disturbance, we would have the diffraction by a hole, whose PSF is determined by the airy pattern.

In the case at hand, however, we do not block the incoming wave and thus we have a non-zero background. This causes the diffracted wave field to oscillate about the background $U_0(x, y; z + z_d)$ based on the distance z_d. This comes straightforwardly from the considerations made to reach Equation (9.3).

We introduce the point disturbance $U_1(x, y)$ at the object plane as

$$U_1(x, y) = \begin{cases} u_1 e^{i\phi_1} - U_0(x, y; z) & if\ (x, y) = (\hat{x}, \hat{y}) \\ 0 & \text{otherwise} \end{cases} \tag{9.5}$$

where $(\hat{x}, \hat{y})$ is the location of the point disturbance, $u_1 \in \mathbb{C}$ and $\phi_1 \in \mathbb{R}$.

In particular, if the object is a pure amplitude object, then there is no phase change, i.e. $\phi_1 = kz + \phi_0$. On the other hand, for a pure phase object, $u_1 = u_0$ and we only have a phase change.

Lenses must be considered explicitly. In fact, most ODT systems rely on magnification and lenses. The expression for the kernel for the defocus with a lens is given by (5.75). This equation is based on the lens-detector distance, but it is valid even if we consider the lens-object distance.

Please note that the convolution of a point disturbance in background-free space with the impulse response is simply the shifted and scaled impulse response. Therefore, Equation (5.75) could be used to compute the point object analytically. However, the integral in (5.75) is not straightforward or easy to compute.

Fortunately, the expression of the focused wave after a lens is known and can be computed by clipping the frequency domain of the wave according to (5.66) if the pupil is known. As circular pupils are common, we will assume that the pupil function is given by Equation (5.71).

Now, we know that we can add any amount of defocus with an arbitrary number of convolutions according to (9.1).

Let $P\colon \mathbb{R}^2 \to \mathbb{R}$ be the circular pupil function in the Fourier domain and $H\colon \mathbb{R}^2 \to \mathbb{C}$ be the transfer function for the angular spectrum method from Equation (5.44).

Then, the simulation of the wave field is given by

$$U(x, y) = \mathcal{F}^{-1}\left\{\mathcal{F}\{U_{Obj}(x, y; z)\}P(u, v)H(u, v; z_d)\right\} \tag{9.6}$$

where $H(u, v; z_d)$ can be either the transfer function of the angular spectrum method or the Fresnel propagator in the frequency domain.

Hereafter, we will always assume that U_{Obj} is the ideal, focused image and therefore drop the parameter z.

For the simulation of the whole acquisition process, we should add a reference wave and phase step it to produce PS-DH and then recover $U(x, y)$. However, this step can be skipped as PS-DH should recover an ideal signal up to round-off errors as it comes from analytical considerations.

The amplitude of $U(x, y)$ can be calculated by taking the absolute value of the wave field, while the phase can be retrieved by unwrapping $\phi_w(x, y) = \arctan 2\left(\Im U(x, y), \Re U(x, y)\right)$.

Please note that during normal ODT reconstruction, we would refocus the wave fields. However, since real extended objects may not lie on a single plane and cannot be refocused completely, there will be no refocusing step during the simulation.

9.1.1 Point Coordinates During Rotation

Now that the model for the diffraction by a point disturbance $U_1(x, y)$ for a single projection is defined, the object must be allowed to rotate about a centre of rotation during the acquisition of the tomographic dataset. Therefore, $\hat{x}$ and z_d vary during acquisition, while $\hat{y}$ is constant, in compliance with the rotation about the y-axis.

To simulate the one-dimensional projection of the point disturbance, we will simulate the two-dimensional projection of the three-dimensional model and then take the line at $y = \hat{y}$. Stacking all the one-dimensional projections of the rotating sample together generates the sinogram of the object.

For simplicity, we put the in-focus plane at the centre of rotation of the CT system and we set the coordinate system for defocus as the one in Fig. 9.1.

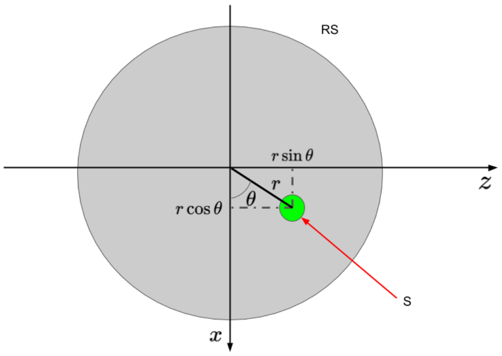


Fig. 9.1: The model for the simulation. The sample (S) is a point disturbance and is placed on the rotating stage (RS). z is the optical axis and the x-axis is parallel to the detector. S is located at $\hat{y}$

The point disturbance is ideally bound to a circumference of radius r that is centred in the centre of rotation, which is now the origin. Therefore, we can model the defocus as the projection onto the optical axis, z, and the position on the detector line as the projection on the x-axis.

For the reconstruction of the phase, we will use FBP. Please note that, in FBP, the contributions of all the projections are summed. This implies that, ideally, all the point on the circumference will contribute to the determination of any point on that circumference, or, more practically, that the spread of the point at a certain distance r in the slice is independent of the initial angle. Therefore, we do not have to generate the sinogram for all the points in the circle of reconstruction, but we can simply generate only the points with initial coordinates $(x_0, z_0) = (r, 0)$ (i.e. $\theta = 0$ at the beginning of the acquisition).

If the detector has N pixels, with $N \in \mathbb{N}$, then we will simulate the distances

$$r_k = k\Delta x \qquad k \in \mathbb{N} \cup \{0\} \;\mid k \leq \frac{N}{2} \tag{9.7}$$

where we have assumed that $x = 0$ is the centre of the detector line, which means that the centre of rotation is at the centre of the slice and FBP does not have to take additional displacement into account.

Please note that point disturbances at different locations will be simulated individually.

Hypothesising constant angular step $\Delta\theta = \frac{2\pi}{S}$, where S is a positive integer that represents the number of projections, then the angle of rotation at the j-th projection is

$$\theta_j = j\Delta\theta \qquad j = 0, 1, ..., S-1 \tag{9.8}$$

Exploiting the initial condition on the coordinates, we have, for the k-th radial distance r_k, $(x_0, z_0)_k = (r_k, 0)$ and, for any projection,

$$x_j = r_k \cos\theta_j \tag{9.9a}$$

$$z_{d;j} = -r_k \sin\theta_j \tag{9.9b}$$

which comes from Equation (4.2).

9.1.2 Relation Between F-Number and Sampling Interval

Remembering the condition for a correct simulation of a lens in (5.72), we can write

$$\frac{f_0}{f_N} \leq \frac{1}{2} \tag{9.10}$$

where f_0 is the cut-off frequency and f_N is the Nyquist frequency as defined by (3.12). This inequality suggests that we should work with the normalised frequencies, i.e. multiply all the frequencies u and v by $\frac{1}{f_N}$ in order to rescale each axis in the Fourier domain to $[-1, 1]$, which means that the true value of the Nyquist frequency is unimportant.

We can explicitly calculate the left-hand side of (9.10) by substituting f_0 with (5.70), which is valid under the the paraxial approximation. Thus, we have

$$\frac{1}{\lambda}\frac{\Delta x}{f/\#} \leq \frac{1}{2} \tag{9.11}$$

where λ is the wavelength of the incident light wave, $f/\#$ is the f-number of the lens and Δx is the sampling interval. We have assumed, for simplicity, that pixel height and pixel width are equal.

Please note that λ in the inequality in (9.11) is usually constant. We are mostly interested to the term $\frac{\Delta x}{f/\#}$. To justify this interest, please consider the transformations

$$f/\#_M = \frac{f/\#}{M} \tag{9.12a}$$

$$\Delta x_M = \frac{\Delta x}{M} \tag{9.12b}$$

for a certain positive real number M. Substitution in (9.11) yields the same result as before since $f/\#$ and Δx are rescaled by the same constant. However, the implication of these substitutions are of the utmost importance. In fact, (9.11) and (9.12) imply that we can either swap the lens and the detector with another set with the same $\frac{\Delta x}{f/\#}$ without affecting the result of the simulation. Or, alternately, change the lens and the magnification and keep the same detector. The result will not change.

This, of course, corroborates the statement made before: we can easily work with normalised frequencies and ideally remove all references to Δx and $f/\#$ as the results depend only on the radius $\frac{f_0}{f_N}$ in the frequency domain. This implies that we could analyse the output of the simulation in the spatial domain by using the pixel as a unit of measurement.

However, we will offer the results in the usual units to offer more physical insight.

9.2 Results

Before starting the simulation, we can compare (5.75) and (9.6) to verify the validity of the latter.

We choose a square detector of side $N = 512$ pixels and sampling interval $\Delta x = \Delta y = 5.5$ µm. We set the wavelength of the incident light wave to $\lambda = 633$ nm. Now we have set the parameters that are easily kept constant, but we have to choose an $f/\#$ that follows the condition in (5.72). We can rearrange the relation to isolate the f-number and obtain

$$f/\# \geq \frac{2\Delta x}{\lambda} \tag{9.13}$$

which puts a constraints on the minimum $f/\#$ allowed in the simulation. Using the chosen parameters, we see that $f/\# \gtrsim 17.378$, so we set $f/\# = 17.4$. In this set-up, the detector side is $L = N\Delta x \approx 2.816$ mm.

For the analytical formula, we also have to find the lens parameter that give the chosen $f/\#$ according to (5.68) and follow the paraxial approximation, which can be interpreted as $\sin\theta \approx \theta$. We choose $w = 3.5$ mm, which entails $z_2 = 2wf/\# \approx 0.1218$ m. We compute the defocus for $S = 1024$

points on the optical axis as $z_{d;i} = i\Delta x$ for $i = 0, 1, ..., S-1$. For the computational formula in (9.6), we will use $d_i = z_{d;i}$, while for the analytical formula in (5.75) we set $d_i = z_2 + z_{d_i}$. Then, we compute the intensity.

Since it is very difficult to compute (5.75), we use an approximation to the analytical formula valid for the optical axis and in presence of small amount of defocus, which is therefore applicable in our case. Indeed, we know that the largest $z_{d;i}$ is $(S-1)\Delta x \approx 5.6265$ mm. Direct comparison shows that this value is much smaller than z_2. Therefore, we can use

$$I(0,0;d_i) = \left(\frac{\pi w^2}{\lambda^2 z_1 z_2}\right)^2 \operatorname{sinc}^2\left(\frac{w^2}{2\lambda z_2}\left(1 - \frac{d_i}{z_2}\right)\right) \tag{9.14}$$

where we used the same notation introduced in (5.63), (5.68) and (5.75).

The normalised results are shown in Fig. 9.2. The two computed intensities seem to be in agreement with each other.

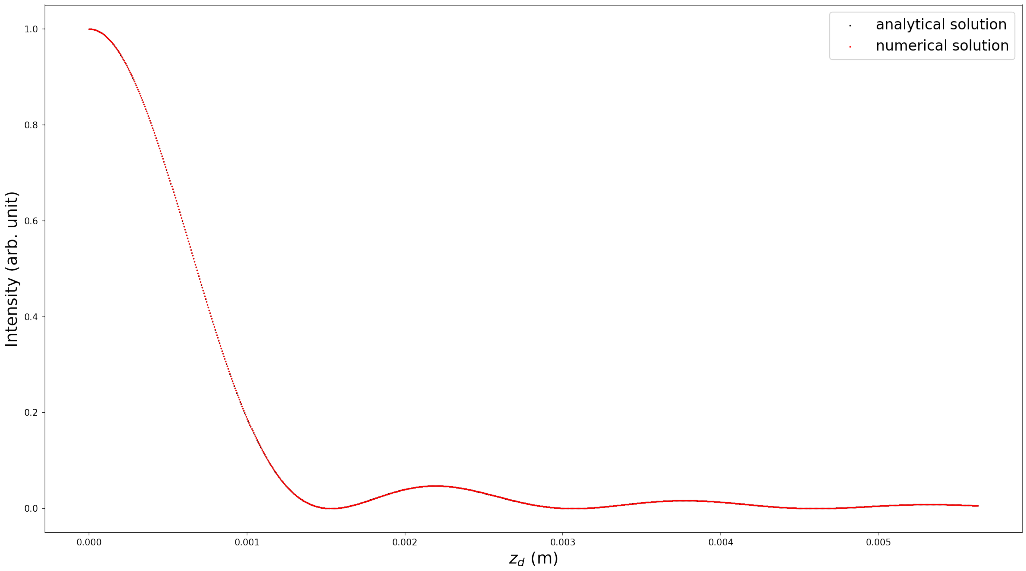


Fig. 9.2: Plot of the numerical (simulated) solution for defocus and the analytical solution on the optical axis. The measured quantity is the intensity.

9.2.1 Comparison of the Pure Phase and Pure Amplitude Models

While it is true that ideally we should not need to use a phase unwrapping routine for the pure phase object it might not always be true. Furthermore, computing the absolute value is usually much faster than finding the (wrapped) phase on a common CPU. Therefore, we are interested in comparing the pure phase and the pure amplitude model.

For the first simulation, we set the pixel width and height to 5.5 µm, the wavelength λ to 633 nm and the $f/\#$ to 17.4. The detector width was 512 pixels, and the height was 128 pixels. We generated 256 initial positions according to (9.7) and acquired 360 projections for each point. Each point was generated along the central line of the detector.

For the amplitude model, when each projection was generated, the central line was extracted and its absolute value was computed and stored into the appropriate sinogram.

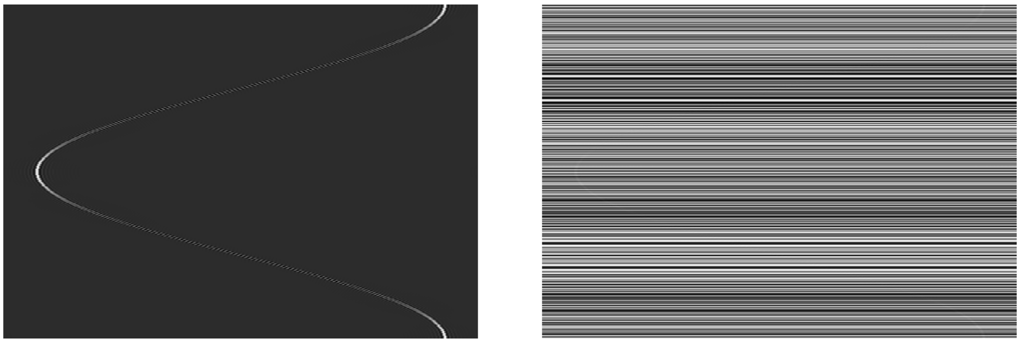
For the phase model, after the wave field was computed, we extracted the phase and unwrapped it, then we extracted the central line and stored it into the sinogram.

All the sinograms were fed to the FBP routine.

Sinogram Comparison

The first step is the sinogram comparison. Here, we expect that the two models will produce remarkably different results. Indeed, in the pure amplitude model the phase is discarded and we only take the absolute value of the amplitude into account. For this reason, we expect a constant uniform background across all the projections.

However, this is usually not true for the pure phase model. This is already hinted by the expression of the background wave (9.2). Since the object wave $U_{Obj}(x, y)$ is computed as interference of the undisturbed background and the diffracted wave (see (9.3)), we expect some variation on the background.

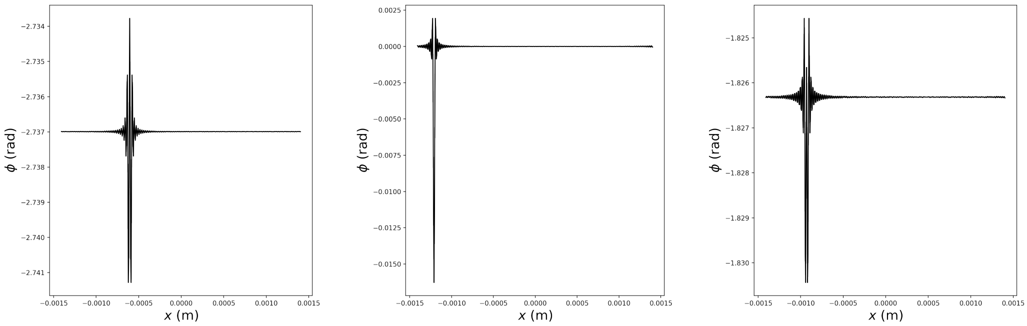


(a) Sinogram for the pure amplitude object with initial position $(x_0, z_0) = (220\Delta x, 0)$.

(b) Sinogram for the pure phase object with initial position $(x_0, z_0) = (220\Delta x, 0)$.

Fig. 9.3: Comparison of the sinograms for the two different models.

According to the results for a point at the initial position $(x_0, z_0) = (220\Delta x, 0)$ in Fig. 9.3, our expectations are satisfied. However, at first sight, it appears that the phase sinogram does not contain any information about the diffracted wave. Fortunately, this is not true. To illustrate this point, we plot the rows number 120, 180 and 220 from 9.3b in Fig. 9.4.



(a) Line profile of the row number 120.

(b) Line profile of the row number 180.

(c) Line profile of the row number 220.

Fig. 9.4: Plots of different rows of the sinogram.

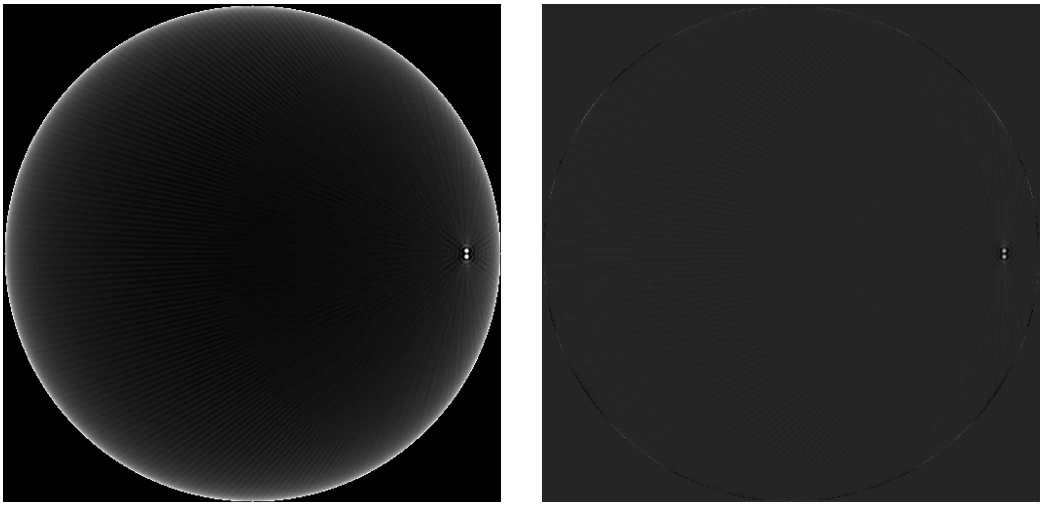
Please note how the background level changes across the projections. The non-sinusoidal background behaviour in the pure phase model is most likely caused by the fact that at each projection the variation of the phase is many times 2π and thus the background wave is not sampled correctly. This is usually not a problem and is rarely observed in experimental ODT since the wave field is refocused. However, background variations are sometimes present even in that case.

The diffracted point behaves similarly in the sinogram of the pure amplitude model. Indeed, the peaks and the wells are clearly visible in Fig. 9.3a. Since this pattern looks periodic and the peak (or the well) is always located on the point coordinates, we expect a "destructive" effect at the true object position when all the projections are combined during FBP. It is of the utmost importance to remark that this splitting may appear only when the maximum defocus is large enough because we have to reach z_d values for which we are near a minimum in the impulse response of the system. Indeed, if the defocus is too small, we would expect to see only one blurred point.

Reconstructed Slices

The slices reconstructed from the sinograms in Fig. 9.3 are shown in Fig. 9.5. The hypothesised split for large radial distances is clearly visible here. Please note that the major difference in the two images lies in the background. In fact, the background in the slice of the amplitude model is minimum at the centre of the slice and slowly increases with the radial distance. On the other hand, the background of the phase model is mostly constant. We would like to emphasise that the background phase was not corrected before the reconstruction and that its disappearance is most likely due to the combination of all the background values, which, assuming phase unwrapping left the phase unchanged, were distributed in $[-\pi, \pi]$.

If we work with the amplitude model, we will show the (reconstructed) absolute amplitude. On the other hand, if we work with the phase model, we will show the (reconstructed) phase.



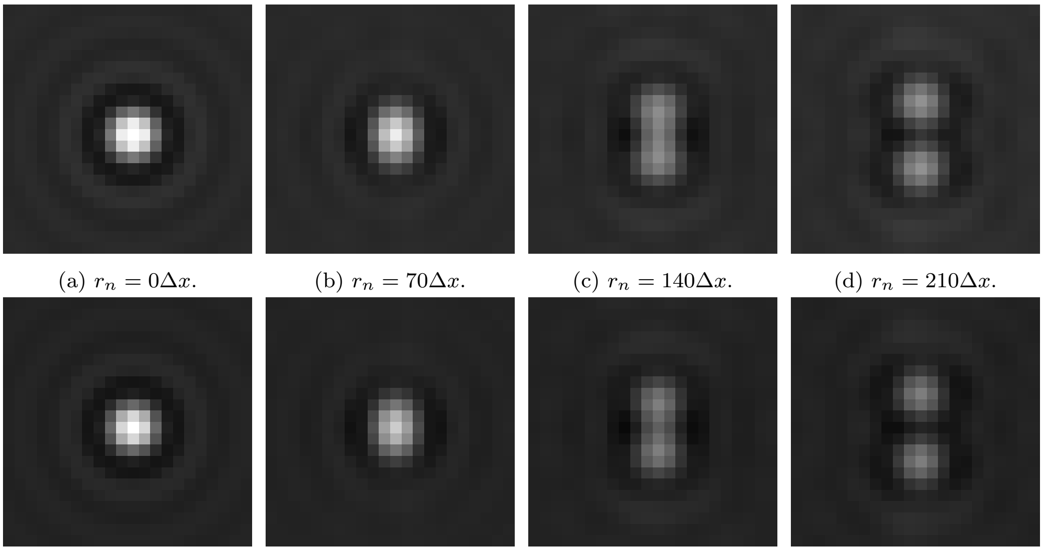
(a) The slice reconstructed from the sinogram of the pure amplitude point at $(x_0, z_0) = (220\Delta x, 0)$.

(b) The slice reconstructed from the sinogram of the pure phase point at $(x_0, z_0) = (220\Delta x, 0)$.

Fig. 9.5: Comparison of the pure amplitude and the pure phase models. Please note that the two point-like objects behave similarly and the major difference lies in the background.

To analyse the models further, we selected a ROI (Region of Interest) centred in the point in four simulated slice for each model. The radial distances of the point are given by $r_n = 70n\Delta x$ for $n = 0, 1, 2, 3$. The ROIs are shown in Fig. 9.6. Even here the two models behave very similarly. The grey level of the point object decreases as its spread and its radial distance from the centre increase. At first sight the spread along the radial direction is approximately constant, while the tangential PSF varies significantly across the image. Therefore, we can say that the PSF of the system is space-variant.

To quantitatively describe the similarities—or dissimilarities—of the two models, we have to compute the FWHM.



(a) $r_n = 0\Delta x$.
(b) $r_n = 70\Delta x$.
(c) $r_n = 140\Delta x$.
(d) $r_n = 210\Delta x$.

(e) $r_n = 0\Delta x$.
(f) $r_n = 70\Delta x$.
(g) $r_n = 140\Delta x$.
(h) $r_n = 210\Delta x$.

Fig. 9.6: Comparison of the PSF at different radial distances. The top row shows the pure amplitude points, while the bottom row contains the results for the pure phase model.

PSF FWHM Analysis

Normally, if the PSF has been either measured or simulated, then the FWHM is estimated through fitting or analytical knowledge of the function. In this case, however, neither solution can be applied.

However, all is not lost. We can analyse the FWHM along the radial and the tangential directions.

Now, if h is a real-valued continuous function that is defined on an interval $[a, b]$ with $a, b, \in \mathbb{R}$ and $a < b$, and if $h(x_1)h(x_2) < 0$ for some x_1 and x_2 such that $a \leq x_1 < x_2 \leq b$, then according to Bolzano's theorem there should be a coordinate $x_s \in [x_1, x_2]$ for which $h(x_s) = 0$. This x_s is the zero-crossing, or root, of h.

We take the line profile of the PSF along either direction. Then, provided we have taken the line passing through the maximum, the PSF will have a background grey level, b, and a peak p. Here $b < p$. Now, the FWHM in the chosen direction is the width of the one-dimensional PSF $h(x)$ at half the maximum height. We can translate this to $\tilde{h}(x) = h(x) - \frac{1}{2}(p - b) = 0$. This means that the coordinates of the points on the PSF that determine the FWHM are roots of $\tilde{h}$ and that we can use a root-finding algorithm to find the FWHM.

Our choice is Newton's method,

$$x_{n+1} = x_n - \frac{\tilde{h}(x_n)}{\tilde{h}'(x_n)} \qquad n \in \mathbb{N} \cup \{0\} \tag{9.15}$$

where $\tilde{h}'$ is the first derivative of $\tilde{h}$ and x_0 is the initial guess for the root, which can be computed as the lower bound to the interval within which we are certain that we have a zero-crossing. In the discrete case, the function is translated by a sequence of $\tilde{h}_i$ such that $\tilde{h}_i = h(x_i)$ and $x_i < x_{i+1}$ for any $i \in \mathbb{N} \cup \{0\}$. Therefore, the set of lower bounds to the roots of h is $\Omega = \{x_i \in \mathbb{R} \mid \ \tilde{h}(x_i)\tilde{h}(x_{i+1}) < 0\}$.

We are interested only on the minimum and the maximum of x_i over Ω because we only have one PSF. Applying Netwon's method to both values will produce two real values, x_l and x_u, such that $x_l < x_u$ and

$$\text{FWHM} = x_u - x_l \tag{9.16}$$

The functions $\tilde{h}$ and $\tilde{h}'$ (9.15) can be calculated with linear or cubic interpolation.

Please note that this method can be used in this case only because noise is negligible. In fact, numerical differentiation of noisy data amplify noise. In some cases, real data may be smoothed with a sliding window (e.g., moving average or any smoothing kernel). However, in that case, the measured FWHM will also depend on the smoothing introduced by the window.

Moreover, we see that the PSF along the radial direction changes the position of the peak, which can be estimated as the argument of the maximum value in the image: i.e. the set of coordinates for which we have a maximum. Unfortunately, this may lead to erroneous calculation of the spread when the splitting is incomplete. Since this is the peak moves along the tangential direction, the FWHM of the tangential PSF is not affected by this effect.

To compute the FWHM, we extracted a 30x30 square ROI centred on the point-like object for each image. Then, we applied the steps described in this section.

For Newton's method, we set the number of iterations to 100 and the tolerance to $t = 10^{-12}$. According to the algorithm used, if x_{n+1} and x_n are more than one tolerance apart after the maximum number of iterations, the routine throws an error; on the other hand, the maximum number of iterations may not be reached. Indeed, the estimated value of the root is returned as soon as the condition on the tolerance is met for the first time.

The results are plotted in Fig. 9.7.

The behaviour of the two models is qualitatively similar. However, they are clearly not identical. Furthermore, the spatial resolution in the pure phase

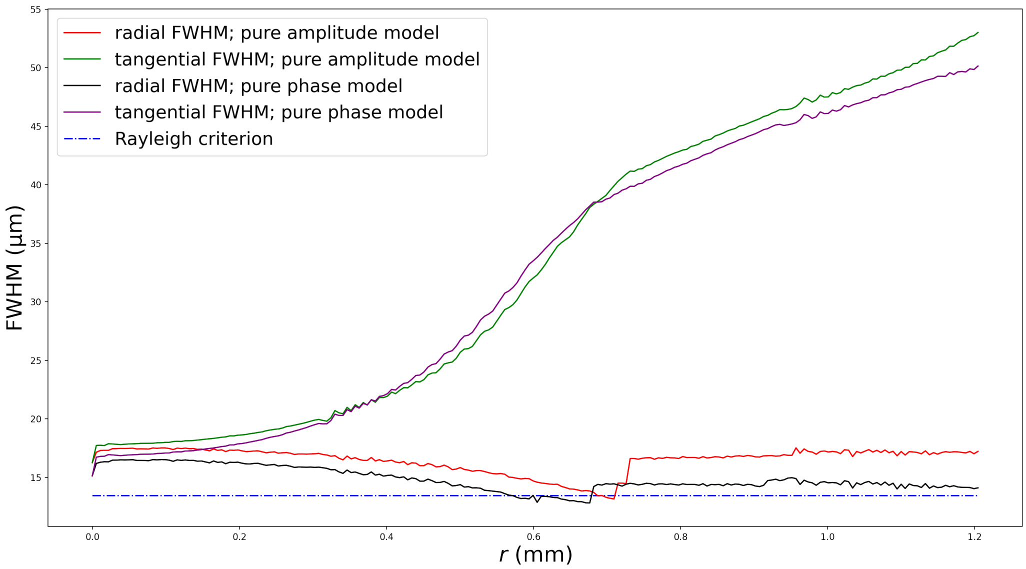
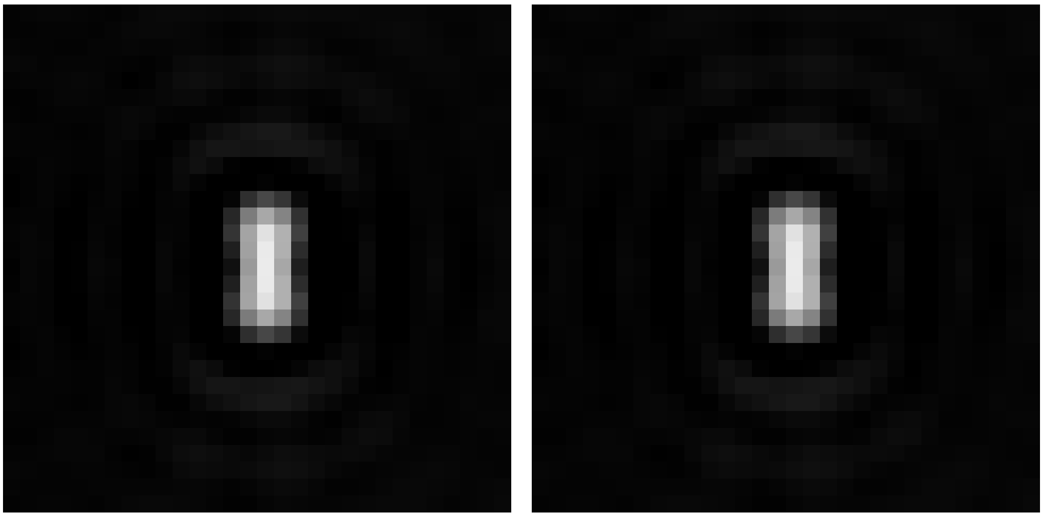


Fig. 9.7: FWHM of the pure phase and the pure amplitude points. While the results for the two models are similar, they are not identical. Moreover, the pure phase model seems to produce better images overall with respect to spatial resolution. In both cases, the radial FWHM is mostly larger than the Rayleigh criterion, which for $f/\# = 17.4$ and $\lambda = 633$ nm is ~ 13.4 µm.



(a) At $r_k = 129\Delta x$, in the amplitude model, we fail to detect the full spread of the radial PSF. We also expect this behaviour for other radial distances r_j near r_k provided $r_j \leq r_k$.

(b) At $r_k = 123\Delta x$, in the phase model, we fail to detect the full spread of the radial PSF. We also expect this behaviour for other radial distances r_j near r_k provided $r_j \leq r_k$.

Fig. 9.8: ROI of the PSF at the (erroneously detected) minimum radial FWHM.

images is better overall. Excluding the dip in the radial FWHM curve, the FWHM is mostly worse than the Rayleigh criterion, which is $\delta = 1.22\lambda f/\# \approx 13.4$ µm. Here, the Rayleigh criterion has been chosen because it is already used to describe the resolution in the projections. Since FBP combines these projections, we expect that the resolution should not be better than the Rayleigh criterion.

To explain the abrupt change in the radial FWHM, we take a look at $r_k = 129\Delta x$ for the amplitude model and $r_k = 123\Delta x$ for the phase model, which are the coordinates for the minima in the radial FWHM curves.

The example in Fig. 9.8 corroborates our assumption. Indeed, in both cases, the peak value is near the waist of the PSF. We therefore caution not to take the results around these two minima as a ground truth as in these cases the spread is ambiguously defined.

The plot in Fig. 9.7 confirms that the PSF is space variant along the tangential direction but seems also to disprove our hypothesis on the radial PSF. In fact, it seems that there might be variations even along this direction. However, due to the inherent problem with the numerical method used to detect the radial FWHM, we prefer not to make any assumption beyond stating that, if it varies, it varies slowly.

Other fluctuations on the results is likely due to the round-off errors introduced by numerical method and finite precision.

9.2.2 Different F-Numbers

In practice, we always try to obtain images with the best possible resolution achievable from the apparatus. In our case, the simulation was limited by the condition (5.72).

However, additional insight might be provided by simulating lenses with larger $f/\#$. Additionally to the simulation already performed, we simulate two other f-numbers. We kept Δx and λ constant and set the other parameters according to Tab. 9.1.

$f/\#$	N (pixels)	M (pixels)	S
25	1024	256	720
35	2048	512	1440

Table 9.1: Parameters for the simulations. N and M are the detector width and height respectively, and S is the number of projections.

For $f/\# = 25$, we simulated all the points according to r_k in Equation (9.7). For $f/\# = 35$, we only simulated a point every 10 pixels.

The Rayleigh criterion for $f/\# = 25$ yields $\delta \approx 19.3$ µm. If $f/\# = 35$, we have $\delta \approx 27.0$ µm.

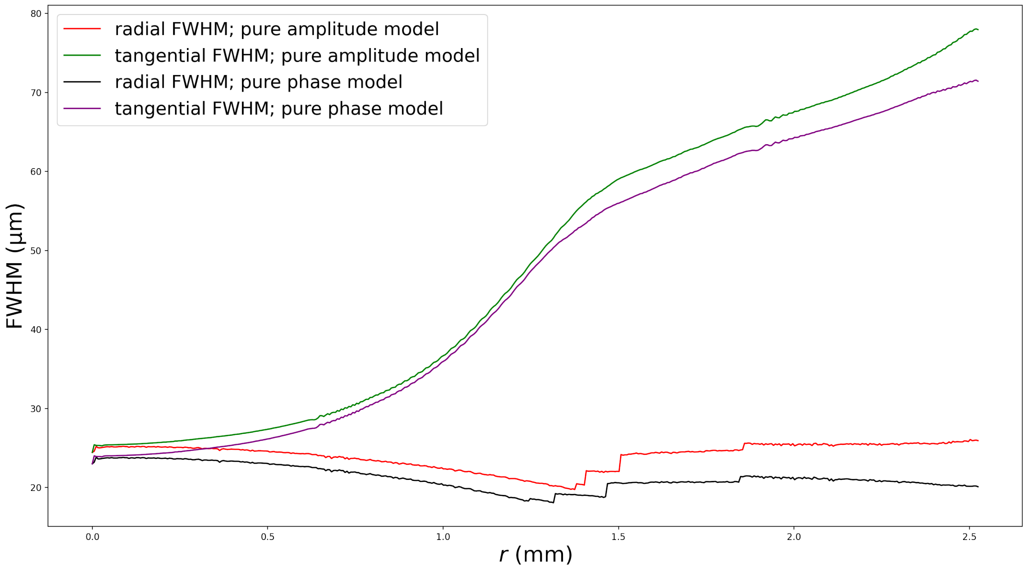


Fig. 9.9: FWHM for $f/\# = 25$.

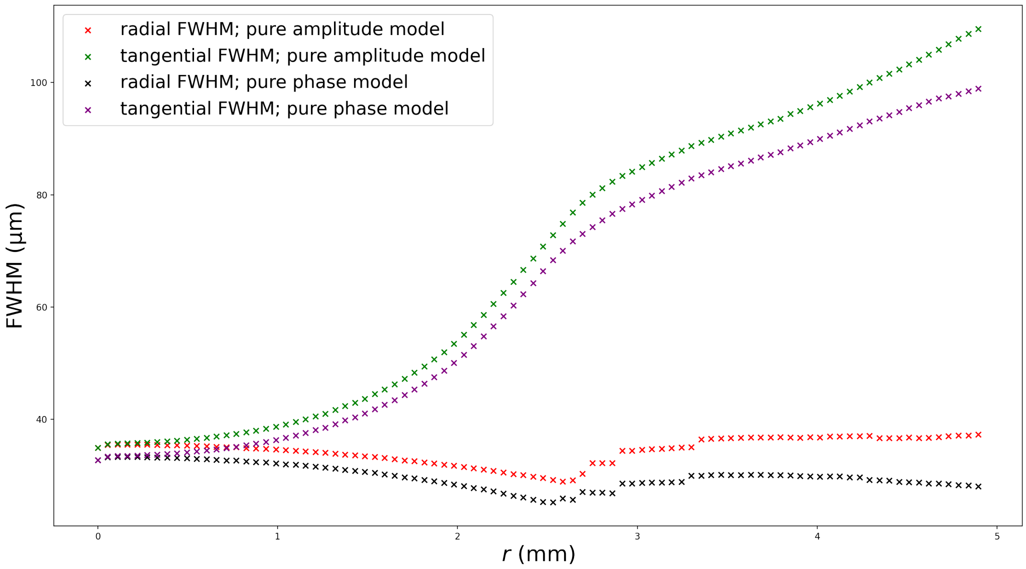


Fig. 9.10: FWHM for $f/\# = 35$.

The FWHM for $f/\# = 25$ is shown in Fig. 9.9, while the FWHM for $f/\# = 35$ is offered in Fig. 9.10. Please note that, aside from the obvious scaling along the r and FWHM axes, the trend at different f-numbers seems mostly identical. This scaling along the radial direction is likely due to the depth of focus, which is proportional to $\frac{1}{\mathrm{NA}^2}$.

Please note that Fig. 9.9 and Fig. 9.10 are offered here to show that, despite the different $f/\#$, the behaviour of the FWHM along the radial and tangential directions is qualitatively similar to the one in Fig. 9.7, although they are rescaled. This is expected and comes from the radius of the circular pupil transfer function in the Fourier domain. Indeed, the smaller the radius, the broader the PSF. For the same reason, a larger defocus distance is required to achieve similar FWHM curves and PSF splitting.

Chapter 10

ODT PSF Deconvolution

The PSF in ODT is space-variant, and therefore the linear imaging system cannot be expressed via the convolution in Equation (3.4), which means that a simple deconvolution routine cannot correct for blurring. However, the PSF along a circumference of radius r is constant. Our system, therefore, has a space-variant PSF with space invariant regions in which it has constant shape but varying orientation.

A transformation of the reconstructed image in polar coordinates produces images in which each row contains information about the radial direction and each column represents the angular coordinate at a fixed radial distance, which in our case, is the tangential direction. Therefore, if we transformed the reconstructed image into polar coordinates, we would generate an image in which each column has its own space invariant tangential PSF.

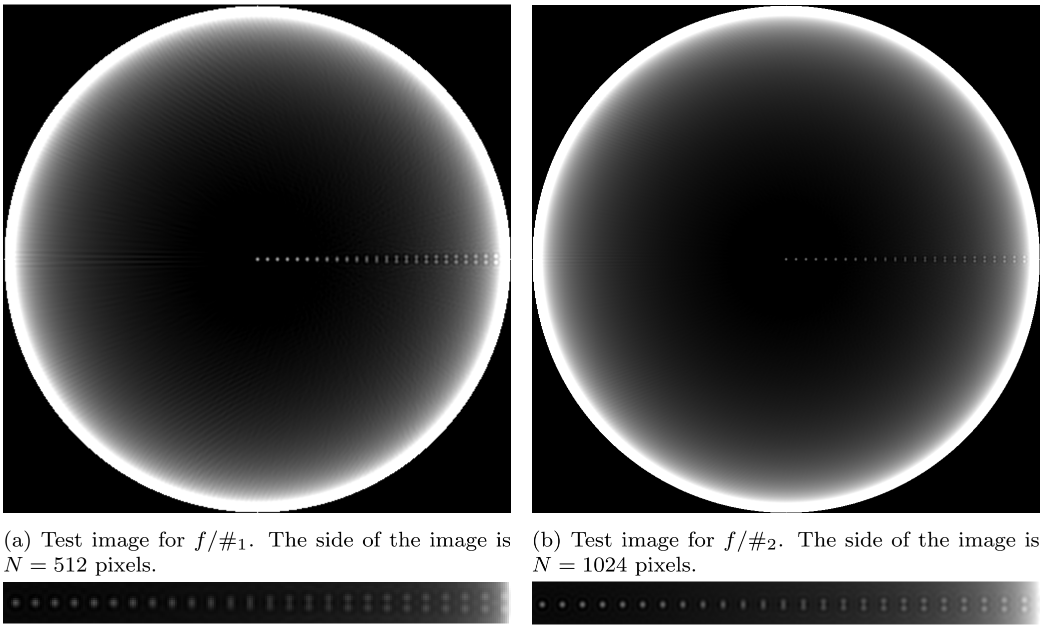
Now, assuming that the radial PSF is roughly space invariant, we can deconvolve the transformed image by using line-by-line one-dimensional deconvolution and revert back to Cartesian coordinates. The deconvolution of each column removes tangential blur, while the deconvolution of each row corrects for the radial PSF.

In some cases, there might be some artefacts on the border of the image in the polar coordinates (r, θ). However, these defects will be located around the central line of the slice in Cartesian coordinates. This is due to the fact that, for $\theta = \pm\pi$, the points in the circle lie on the central line in the left half of the image.

Thankfully, we can easily remove this kind of artefact. Let $I(x, y)$ be the image in Cartesian coordinates. To fix this problem, we also generate an image by inverting the x-axis in $I(x, y)$, i.e. $\tilde{I}(x, y) = I(-x, y)$. Now, transforming both images into polar coordinates and deconvolving before reverting them back to Cartesian coordinates generates two images with the same type of artefact, but in different regions since the inversion of the x-axis introduced a shift in the angular coordinate. For simplicity, now the same notation can be used for the deconvolved images. Now we invert the x-axis of $\tilde{I}(x,y)$ again and call the new image $A(x,y)$. Summing the left half of $A(x,y)$ and the right half of $I(x,y)$ completely removes the artefact as we are only keeping the unaffected regions.

To improve the results of image restoration, if N is the side of the square slice, then, in the polar transformed image, the number of pixel on the radial direction is $R = \frac{N}{2}$ and on the tangential direction we have $M = 4N$. This is because of the behaviour of the arc length with respect to the radius of the circumference, as shown in Equation (4.5) and Fig. 4.3.

10.1 Pure Amplitude Model



(a) Test image for $f/\#_1$. The side of the image is $N = 512$ pixels.

(b) Test image for $f/\#_2$. The side of the image is $N = 1024$ pixels.

(c) ROI of the test image in Fig. 10.1a.

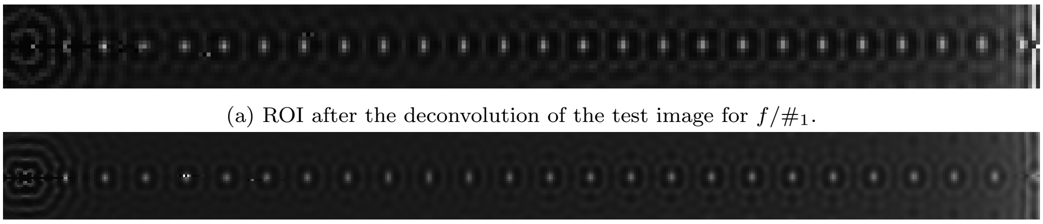
(d) ROI of the test image in Fig. 10.1b.

Fig. 10.1: The top row shows the test images used, while the bottom row shows the respective ROI that contains all the points. Please note that the image in Fig. 10.1a is four times as small as Fig. 10.1b.

For the pure amplitude model, we constructed two images: one for $f/\#_1 = 17.4$ and one for $f/\#_2 = 25$.

In the first image, we took the simulated images for a point located on the x-axis at $x_k = r_k = 10k\Delta x$—with $r_k \leq N\Delta x$—and we summed these images together. In the second, we used $x_k = 20k\Delta x$ and $x_k \leq N\Delta x$ instead, but we left the procedure unvaried.

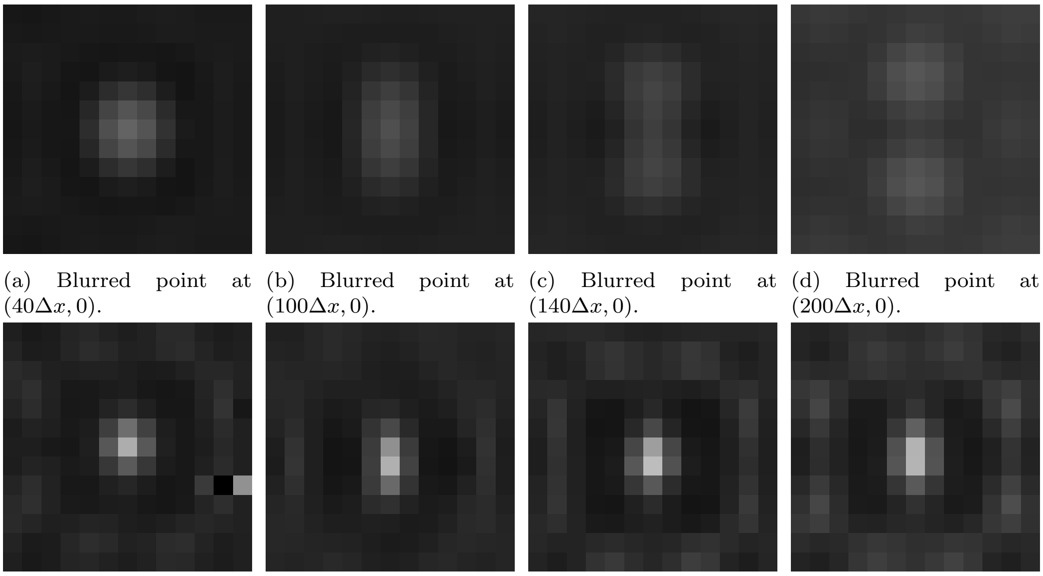
The test images are shown in Fig. 10.1. Please note that the images look the same size, but they have been rescaled to fit in the figure. In fact, Fig. 10.1a is four times as small as Fig. 10.1b.



(a) ROI after the deconvolution of the test image for $f/\#_1$.

(b) ROI after the deconvolution of the test image for $f/\#_2$.

Fig. 10.2: The same ROIs used in Fig. 10.1, but taken from the deconvolved images. Again, the images have been rescaled to fit the page. Indeed, Fig. 10.2a is smaller than Fig. 10.2b by a factor of 2 on both sides.



(a) Blurred point at $(40\Delta x, 0)$.

(b) Blurred point at $(100\Delta x, 0)$.

(c) Blurred point at $(140\Delta x, 0)$.

(d) Blurred point at $(200\Delta x, 0)$.

(e) Deconvolved point at $(40\Delta x, 0)$.

(f) Deconvolved point at $(100\Delta x, 0)$.

(g) Deconvolved point at$(140\Delta x, 0)$.

(h) Deconvolved point at $(200\Delta x, 0)$.

Fig. 10.3: ROIs extracted from the test image generated for $f/\#_1 = 17.4$. The top row shows the blurred points, while the bottom row offers the results after deconvolution.

Deconvolution was performed on both the tangential and the radial directions by using Richardson-Lucy's deconvolution algorithm for the one-dimensional case. The number of iterations for the tangential direction was set to 150, while for the radial direction we only did 50 iterations.

The results—shown only for the ROIs—is offered in Fig. 10.2, after being corrected for the outliers. The deconvolution algorithm introduces ripple-like fluctuations near the centre of the slice (left side in images in Fig. 10.2a). Furthermore, each point object is surrounded by a ring that is darker than the background. While Richardson-Lucy's algorithm introduces artefacts in the images, it also decreases blurring significantly on both the tangential and the radial directions.

To offer a fairer comparison, we take four 13x13 ROIs centred in the point at 40, 100, 140 and 200 pixels from the centre of the slice in all the images. The comparison of the simulated points and the results of the deconvolution routine for $f/\#_1$ is shown in Fig. 10.3. Please note that the spatial resolution has not been brought down to a single pixel, although blurring is significantly reduced along both directions. Nonetheless, the spread along the tangential direction is still, in general, worse than that along the radial direction.

Since radial blurring is visibly less severe than the one in the tangential direction, we offer a visual comparison of the line profile of the central line in the selected ROI (Fig. 10.1c and 10.2a). In these ROIs, the point starts to split at the radial distance $r_s \approx 130\Delta x$. Since we have a point every 10 pixels, we decided to limit the line profile to the region for which $r \leq 135\Delta x$ in order to show the point at r_s.

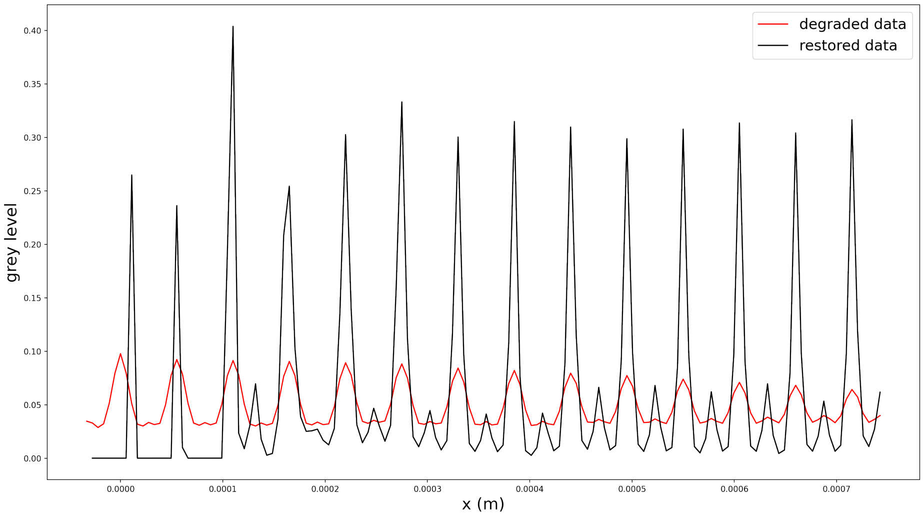


Fig. 10.4: Radial line profile for $r \leq 135\Delta x$. Please note that the restored peaks are higher and narrower than the blurred ones. The line profile of the restore data was misaligned by one pixel and has been shifted to compare the pixels.

The line profile is shown in Fig. 10.4. Here the restored peaks have clearly higher grey levels and are narrower than their blurred counterpart. Furthermore, although it is more visible in Fig. 10.1c and 10.3, even here it is possible to see that the grey levels of the restored peaks does not decreases as the radial distance from the centre of the slice increases as it was the case for the degraded image.

The restored peaks for $r = 0\Delta x$ and $r = 10\Delta x$ are problematic; in fact, direct comparison with Fig. 10.3 shows that they could easily be confused with—and may be—artefacts.

In Fig. 10.5 are shown the line profiles for the blurred and restored points shown in Fig. 10.3. Please note the shape of the restored points in comparison with the original PSFs.

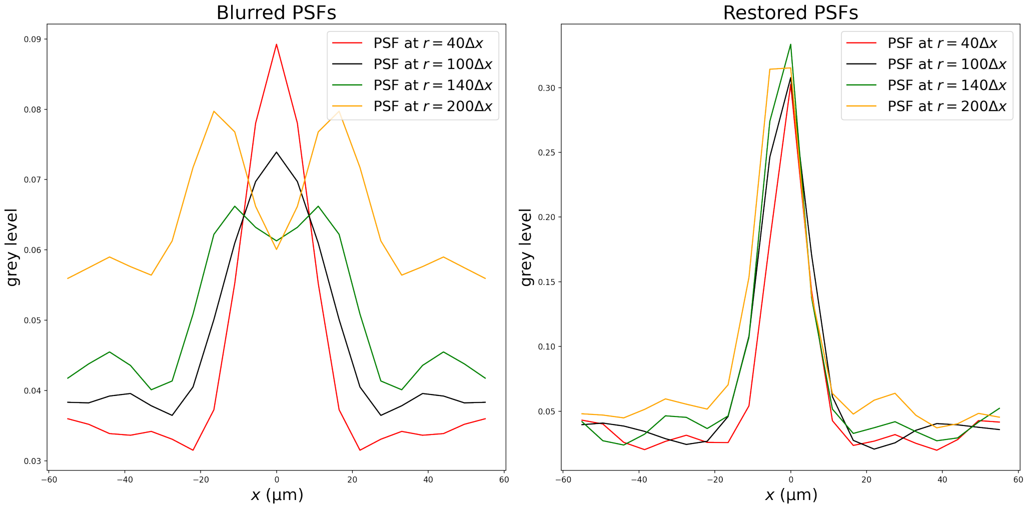
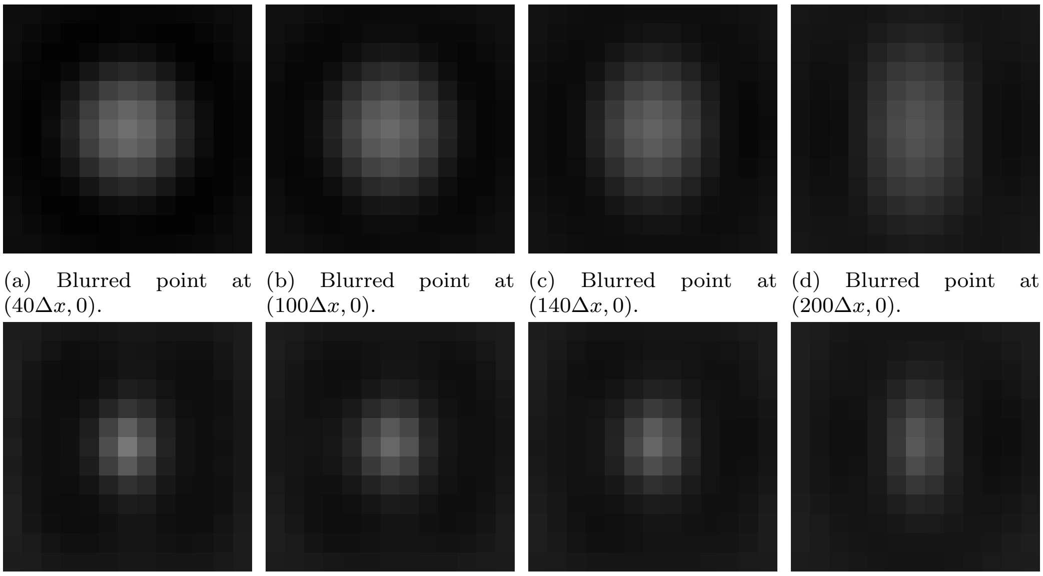


Fig. 10.5: Tangential blurring from Fig. 10.3: on the left, we have the tangential PSFs for the degraded points, while on the right we have the tangential PSFs after restoration.

Unfortunately, the region around the centre of the slice, which is on the left side of the ROI, could not be corrected properly. This might be due to the very low number of samples along the circumferences at low radial distance in the original image.

The comparison of the ROIs for $f/\#_2$ is shown in Fig. 10.6. Please note that even in this case the blur is significantly reduced, although it remains consistently worse than the one in the bottom row of Fig. 10.3. We suspect that, if the initial image is too degraded and has lost most of the information about the high spatial frequencies during image formation, complete restoration may not be possible with common algorithms like Richardson-Lucy's.



(a) Blurred point at $(40\Delta x, 0)$.

(b) Blurred point at $(100\Delta x, 0)$.

(c) Blurred point at $(140\Delta x, 0)$.

(d) Blurred point at $(200\Delta x, 0)$.

(e) Deconvolved point at $(40\Delta x, 0)$.

(f) Deconvolved point at $(100\Delta x, 0)$.

(g) Deconvolved point at$(140\Delta x, 0)$.

(h) Deconvolved point at $(200\Delta x, 0)$.

Fig. 10.6: ROIs extracted from the test image generated for $f/\#_2 = 25$. The top row shows the blurred points, while the bottom row offers the results after deconvolution.

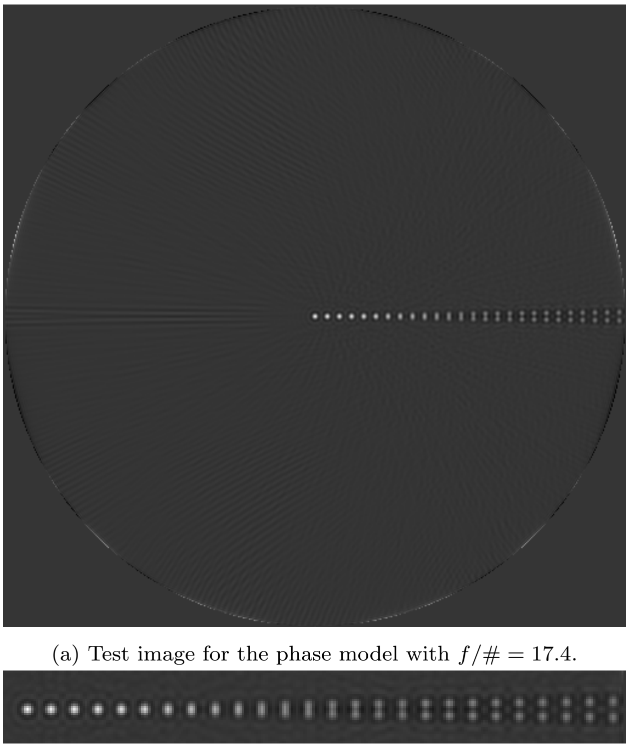
10.2 Pure Phase Model

For the pure phase model, we only tested a single image composed by the sum of the points with the same strategy described for $f/\#_1$ in the pure amplitude model. Even in this case, the f-number was $f/\# = 17.4$. The test image, alongside the ROI, is shown in Fig. 10.7.

Line-by-line deconvolution via Richardson-Lucy's algorithm failed regardless of the number of iterations during the test, even when we tried to skip the radial direction, so we decided to use a one-dimensional Wiener deconvolution routine in polar coordinates.

The results of the line-by-line deconvolution along the tangential direction is shown in Fig. 10.8. Please note that the spread along the radial direction remains unchanged because we did not correct for radial blur. On the other hand, tangential blur is significantly reduced. The point at $r_k = 0$ pixels from the centre of the slice could not be restored and generated a NaN, so we set it to 0 and ignored it. This is not a problem when the object is placed off-centre during acquisition, which is recommended to improve the SNR [47].

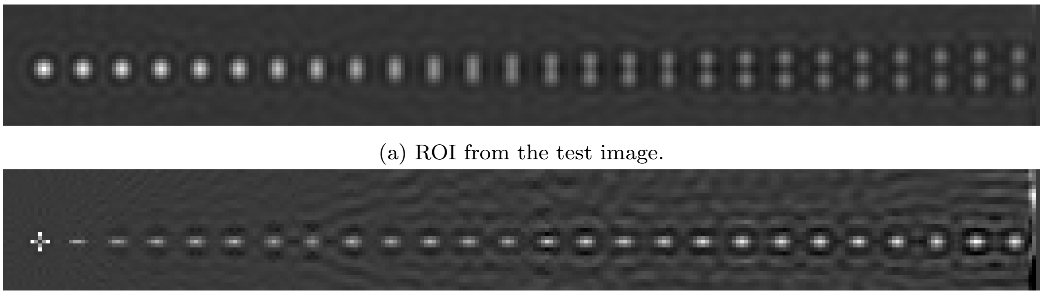
Unfortunately, even with Wiener deconvolution, correction for radial blur failed to produce good results. This may be due to the errors introduced in



(a) Test image for the phase model with $f/\# = 17.4$.

(b) ROI that contains the point objects in the test image.

Fig. 10.7: Test image (top) and ROI (bottom) for the pure phase model with $f/\# = 17.4$.



(a) ROI from the test image.

(b) ROI after restoration.

Fig. 10.8: The ROI from the test image (top) and the restored ROI (bottom) via one-dimensional Wiener deconvolution. The top image is the same as the one provided in Fig. 10.7.

previous steps or to the suboptimal value of the tuning parameter for the deconvolution algorithm.

Chapter 11

Conclusion

The reconstruction software was optimised by lifting some of the most stringent limitations in the old script and introducing more flexibility at the cost of larger storage requirements. Moreover, moderate speed-up has been observed.

Resolution in ODT was modelled and simulated with the aid of computational Fourier optics strategies for scalar diffraction by a point-like objects. Spatial resolution was assessed qualitatively and quantitatively by using numerical routines to estimate the FWHM for each point and was verified to be space variant with space invariant regions in both the pure phase model and the pure amplitude one. Their comparison shows that, while similar, these two models are not identical.

The models are simple and flexible. For this reason, they can easily be adapted to a real case scenario to generate ideal PSFs for the deconvolution of real datasets, which—if successful—will allow for more reliable spatial measurements of the specimens.

Image restoration was used to correct for blur. For this part, line-by-line deconvolution in polar coordinates was chosen because it offered a simpler definition of the regions regions with space invariant PSFs. Simple deconvolution routines show that the tangential blur is greatly improved during this step, although the centre of the image cannot be determined. In general, this is not not a problem as it is preferred to place the specimen off-centre because it improves the SNR.

Unfortunately, while there is great improvement, the spatial resolution of the reconstructed slices did not reach the size of a single pixel. This is particularly true if the acquisition parameters are not close to the diffraction limit. In fact, deconvolution of test images obtained for large f-numbers showed that blur was still evident. This is likely caused by the lack of frequency content in the slice caused by the pupil transfer function.

Future developments could lead to experimental works either to prove or to disprove this model. Another option could be the study of more sophisticated algorithms for optimal image deconvolution. Should these steps succeed, ODT will become suitable for the acquisition and quantitative analysis of samples that are either not perfectly focused or too large to be focused. Furthermore, the accuracy of the model for ODT of the PSF with the Mach-Zehnder interferometer may be improved by considering noise and detector efficiency, introducing the effect of the other components in the Mach-Zehnder interferometer and including aberrations.

Bibliography

[1] Serge Lang. *Introduction to Linear Algebra*. Undergraduate Texts in Mathematics. Springer New York, New York, NY, 1986.

[2] Joseph W. Goodman. *Introduction to Fourier Optics*. McGraw Hill, 2nd edition, 1996.

[3] Richard E. Gonzalez, Rafael C.; Woods. *Digital Image Processing*. Pearson, 4th edition, 2018.

[4] Daisuke Takahashi. *Fast Fourier Transform Algorithms for Parallel Computers*, volume 2 of *High-Performance Computing Series*. Springer Singapore, Singapore, 2019.

[5] William Hadley Richardson. Bayesian-Based Iterative Method of Image Restoration. *Journal of the Optical Society of America*, 62(1):55, jan 1972.

[6] Li Xu, Jimmy S.J. Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. *Advances in Neural Information Processing Systems*, 2(January):1790–1798, 2014.

[7] B. R. Hunt. The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer. *IEEE Transactions on Computers*, C-22(9):805–812, 1973.

[8] Deepa Kundur and Dlmltrlos Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, may 1996.

[9] S Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, may 1983.

[10] Ke-Lin Du and M. N. S. Swamy. Simulated Annealing. In *Search and Optimization by Metaheuristics*, pages 29–36. Springer International Publishing, Cham, 2016.

[11] B. C. McCallum. Blind deconvolution by simulated annealing. *Optics Communications*, 75(2):101–105, 1990.

[12] D. A. Fish, J. G. Walker, A. M. Brinicombe, and E. R. Pike. Blind deconvolution by means of the Richardson–Lucy algorithm. *Journal of the Optical Society of America A*, 12(1):58–65, 1995.

[13] James G. Nagy and Dianne P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM Journal of Scientific Computing*, 19(4):1063–1082, 1998.

[14] Tod Lauer. Deconvolution with a spatially-variant PSF. *Astronomical Data Analysis II*, 4847(520):167, 2002.

[15] Avinash C. Kak and Malcolm Slaney. 3. Algorithms for Reconstruction with Nondiffracting Sources. In *Principles of Computerized Tomographic Imaging*, pages 49–112. Society for Industrial and Applied Mathematics, jan 2001.

[16] Stephen G. Azevedo, Daniel J. Schneberk, J. Patrick Fitch, and Harry E. Martz. Calculation of the Rotational Centers in Computed Tomography Sinograms. *IEEE Transactions on Nuclear Science*, 37(4):1525–1540, 1990.

[17] Nghia T. Vo, Michael Drakopoulos, Robert C. Atwood, and Christina Reinhard. Reliable method for calculating the center of rotation in parallel-beam tomography. *Optics Express*, 22(16):19078, 2014.

[18] Grant T Gullberg, Carl R Crawford, and Benjamin M W Tsui. Reconstruction Algorithm for Fan Beam with a Displaced Center-of-Rotation. *IEEE Transactions on Medical Imaging*, 5(1):23–29, mar 1986.

[19] Avinash C. Kak and Malcolm Slaney. 6. Tomographic Imaging with Diffracting Sources. In *Principles of Computerized Tomographic Imaging*, pages 203–273. Society for Industrial and Applied Mathematics, jan 2001.

[20] S. Pan and A. Kak. A computational study of reconstruction algorithms for diffraction tomography: Interpolation versus filtered-backpropagation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(5):1262–1275, oct 1983.

[21] Ulf Schnars, Claas Falldorf, John Watson, and Werner Jüptner. *Digital Holography and Wavefront Sensing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[22] David G. Voelz. *Computational Fourier Optics: A MATLAB Tutorial*. SPIE, 2011.

[23] Yafei Xing, Mounir Kaaniche, Béatrice Pesquet-Popescu, and Frédéric Dufaux. Fundamental Principles of Digital Holography. In *Digital Holographic Data Representation and Compression*, chapter 2, pages 3–18. Elsevier, 2016.

[24] Alessandro Bettini. *A Course in Classical Physics 4 - Waves and Light*. Undergraduate Lecture Notes in Physics. Springer International Publishing, Cham, 2017.

[25] J. W. Goodman. Digital Image Formation From Electronically Detected Holograms. In *Computerized Imaging Techniques*, volume 0010, pages 176–181, jul 1967.

[26] M. Kronrod, N. Merzlyakov, and L. Yaroslavskii. Reconstruction of a Hologram with a Computer. *Soviet Physics Technical Physics*, 17:333–334, 1972.

[27] U. Schnars and W. Jüptner. Direct recording of holograms by a CCD target and numerical reconstruction. *Applied Optics*, 33(2):179–181, 1994.

[28] Emmett N. Leith and Juris Upatnieks. Wavefront Reconstruction with Diffused Illumination and Three-Dimensional Objects. *Journal of the Optical Society of America*, 54(11):1295–1301, nov 1964.

[29] Gu-Liang Chen, Ching-Yang Lin, Ming-Kuei Kuo, and Chi-Ching Chang. Numerical suppression of zero-order image in digital holography. *Optics Express*, 15(14):8851–8856, 2007.

[30] Chandra Sekhar Seelamantula, Nicolas Pavillon, Christian Depeursinge, and Michael Unser. Zero-order-free image reconstruction in digital holographic microscopy. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 201–204. IEEE, jun 2009.

[31] Ichirou Yamaguchi and Tong Zhang. Phase-shifting digital holography. *Optics Letters*, 22(16):1268–1270, aug 1997.

[32] Ichirou Yamaguchi, Jun-ichi Kato, Sohgo Ohta, and Jun Mizuno. Image formation in phase-shifting digital holography and applications to microscopy. *Applied Optics*, 40(34):6177–6186, 2001.

[33] Etienne Cuche, Frédéric Bevilacqua, and Christian Depeursinge. Digital holography for quantitative phase-contrast imaging. *Optics Letters*, 24(5):291–293, 1999.

[34] Etienne Cuche, Pierre Marquet, and Christian Depeursinge. Simultaneous amplitude-contrast and quantitative phase-contrast microscopy by numerical reconstruction of Fresnel off-axis holograms. *Applied Optics*, 38(34):6994–7001, 1999.

[35] J M Huntley. Noise-immune phase unwrapping algorithm. *Applied Optics*, 28(16):3268–3270, aug 1989.

[36] Dennis C. Ghiglia and Louis A. Romero. Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods. *Journal of the Optical Society of America A*, 11(1):107–117, 1994.

[37] Mark D. Pritt. Comparison of path-following and least-squares phase unwrapping algorithms. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2:872–874, 1997.

[38] Miguel Arevallilo Herráez, David R. Burton, Michael J. Lalor, and Munther A. Gdeisat. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Applied Optics*, 41(35):7437–7444, 2002.

[39] Tatiana Latychevskaia and Hans-werner Fink. Practical algorithms for simulation and reconstruction of digital in-line holograms. *Applied Optics*, 54(9):2424, 2015.

[40] Pablo Piedrahita-quintero and Jorge Garcia-sucerquia. Off-axis digital holography simulation in ImageJ. *Optik - International Journal for Light and Electron Optics*, 140:626–633, 2017.

[41] Yongjin Sung, Wonshik Choi, Christopher Fang-Yen, Kamran Badizadegan, Ramachandra R Dasari, and Michael S Feld. Optical diffraction tomography for high resolution live cell imaging. *Optics Express*, 17(1):266–277, jan 2009.

[42] Kyoohyun Kim, Jonghee Yoon, Seungwoo Shin, Sangyun Lee, Su-A Yang, and Yongkeun Park. Optical diffraction tomography techniques for the study of cell pathophysiology. *Journal of Biomedical Photonics & Engineering*, pages 020201–1–020201–16, 2016.

[43] Kyoohyun Kim, Jonghee Yoon, and YongKeun Park. Large-scale optical diffraction tomography for inspection of optical plastic lenses. *Optics Letters*, 41(5):934, 2016.

[44] Jos van Rooij and Jeroen Kalkman. Large-scale high-sensitivity optical diffraction tomography of zebrafish. *Biomedical Optics Express*, 10(4):1782–1973, 2019.

[45] Jos van Rooij and Jeroen Kalkman. Polarization contrast optical diffraction tomography. *Biomedical Optics Express*, 11(4):2109, 2020.

[46] Paul Müller, Mirjam Schürmann, and Jochen Guck. ODTbrain: A Python library for full-view, dense diffraction tomography. *BMC Bioinformatics*, 16(1):1–9, 2015.

[47] Julianna Kostencka, Tomasz Kozacki, Michał Dudek, and Małgorzata Kujawińska. Noise suppressed optical diffraction tomography with autofocus correction. *Optics Express*, 22(5):5731, 2014.

[48] Myung K. Kim. *Digital Holographic Microscopy*, volume 162 of *Springer Series in Optical Sciences*. Springer New York, New York, NY, 2011.

[49] B. Recur, J. P. Guillet, I. Manek-Hönninger, J. C. Delagnes, W. Benharbone, P. Desbarats, J. P. Domenger, L. Canioni, and P. Mounaix. Propagation beam consideration for 3D THz computed tomography. *Optics Express*, 20(6):5817–5829, 2012.

[50] Anna K. Trull, Jelle Van Der Horst, Willem Jan Palenstijn, Lucas J. Van Vliet, Tristan Van Leeuwen, and Jeroen Kalkman. Point spread function based image reconstruction in optical projection tomography. *Physics in Medicine and Biology*, 62(19):7784–7797, 2017.

[51] Jelle van der Horst and Jeroen Kalkman. Image resolution and deconvolution in optical tomography. *Optics Express*, 24(21):24460, 2016.