Stony Brook University

Computationally Efficient Methods for Shift-variant Image Restoration in Two and Three Dimensions

A Dissertation Presented by

Shekhar Bangalore Sastry

to

The Graduate School

In Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

December 2011

Stony Brook University

The Graduate School

Shekhar Bangalore Sastry

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Dr. Muralidhara Subbarao – Dissertation Advisor
Professor, Electrical and Computer Engineering

Dr. Petar M. Djuric – Chairperson of Defense
Professor, Electrical and Computer Engineering

Dr. Monica F. Bugallo
Assistant Professor, Electrical and Computer Engineering

Dr. Klaus Mueller
Associate Professor, Computer Science

This dissertation is accepted by the Graduate School

Lawrence Martin

Dean of the Graduate School

Abstract of the Dissertation

Computationally Efficient Methods for Shift-variant Image Restoration in Two and Three Dimensions

by

Shekhar Bangalore Sastry

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

2011

Shift-variant image restoration or image deblurring is useful in many applications including Machine Vision, Image Processing, 3D Microscopy, medical image analysis, etc. Currently, several shift-variant restoration approaches exist. However, they are either computationally expensive or inaccurate leading to poor image quality. This thesis proposes and investigates computationally efficient techniques that produce high quality restoration, even in the presence of noise. The methods presented here are general in that they are not limited to certain types of kernels to be computationally efficient. Detailed analysis and computational algorithms for implementing the methods are provided.

This thesis addresses blurring in linear shift-variant imaging systems in both two and three dimensions. Image restoration in such systems corresponds to solving the

Fredholm Integral Equation of the Fist Kind. In the two dimensional case, computational efficiency is achieved through *localization*. In the case of three dimensions, a new domain transformation is applied to achieve computational efficiency. These results are presented in two parts.

In the first part, three image restoration algorithms are discussed. The first algorithm is a localized approach to restore highly defocused images. It is based on an existing method called the *single-interval RT (SRT)* method. The SRT method is found to restore only small to medium levels of blur. It is extended to restore images blurred with large shift-variant point spread functions (PSFs). The new method is called the *multi-interval RT (MRT)* method. In the MRT technique, the region around a pixel, with size comparable to the support domain of the blurring kernel, is divided into several smaller regions (intervals). The blurred image in each interval is modeled separately by truncated Taylor-series polynomials. A linear system is derived by differentiating the polynomial with respect to spatial variables. A vector of blurred image derivatives is then expressed as sum of such linear systems. An iterative update formula is obtained that is evaluated to improve the focused image estimate. Experimental results for the MRT technique in 1D on analytic functions and in 2D on simulation data and real images are presented. The results show that MRT technique is effective for restoring highly defocused images but at a modest increase in computation cost compared to SRT.

The next two restoration algorithms are iterative versions of SRT. One of them is the RT Iterative (RTI) method. In the RTI method, forward RT equation (of SRT) which expresses the blurred image as a weighted sum of focused image and its derivatives is rearranged to form an update equation. The RTI update equation is found to converge rapidly to a solution. The other method is a modification of the gradient based Landweber's iteration and is called the RT based Landweber's (RTLW) algorithm. The RTLW algorithm has a step-size parameter and hence provides more control over the convergence to the solution. Both RTI and RTLW methods are analyzed for computational complexity. It is found that for deblurring defocus aberration, the RTI and RTLW methods are $O(NlogN)$ complex per iteration. Both the methods are compared with Landweber's algorithm and Tikhonov regularization (using SVD), for computation time, accuracy, robustness against noise and quality of restored images. An interesting new insight towards ill-conditioned nature of the image restoration problem becomes apparent by analyzing the localized methods.

The second part of this thesis focuses on a new theorem called the Generalized Convolution Theorem (GCT). GCT provides the conditions under which a linear shift-variant system could be transformed to a linear shift-invariant system. The motivation

for such transformation is the computational advantage of implementing shift-invariant systems and shift-invariant deblurring using the Fast Fourier Transform (FFT). In the transformed domain the shift-invariant equivalent of a shift-variant system is deblurred in $O(NlogN)$. Implementing the transformations is not computationally expensive. Hence, shift-variant restoration becomes computationally efficient. GCT is stated and proved in one dimension (1D). The 1D GCT is applied to a hypothetical imaging system for verification. A proof of multi-dimensional version of GCT is also provided.

Next, applications of GCT in 3D imaging with digital cameras and microscopes are considered. Blurred 3D image sequence is modeled as the result of shift-variant filtering with a 3D PSF. It is found that the 3D shift-variant kernel under geometric optics satisfies the conditions required by GCT for domain transformation. Therefore, GCT is applied to 3D deconvolution microscopy. Specifically, GCT is useful in reducing computational requirements of shift-variant or depth-dependent deconvolution techniques. Simulation experiments in 3D compare GCT with shift-invariance (SI) approximation and piecewise constant shift-invariance (PCSI) approximations. It is demonstrated that GCT provides better results both qualitatively and quantitatively when compared to SI and PCSI approximations. Moreover, GCT is also found to mitigate some of the artifacts common in deconvolution microscopy. Shape recovery using GCT is also briefly investigated.

To my parents

Table of Contents

List of figures

List of tables

List of algorithms

Acknowledgements

In here, I would like to express my deep gratitude and thank the people who have helped me directly and indirectly during this journey. First of all many thanks go to my adviser Prof. Murali Subbarao, especially for being patient with me through my mistakes and for providing me with the freedom to pursue the research of my interest at my own pace. The encouragement and support I received even for naïve ideas have given me the confidence to try new things. I am also grateful for his kindness and support. I would like to thank Prof. Petar M. Djurić, Prof. Mónica F. Bugallo and Prof. Klaus Müller for agreeing to serve on my dissertation committee. Their insightful comments have greatly helped shape this work.

My colleagues, Dr. Xue Tu and Dr. Youn-sik Kang, helped me get acclimatized to the research environment. Some of their work has been used in my research and I thank them for that and for their friendship. I would also like to thank Dr. Satyaki Dutta for the many lengthy phone call discussions; they helped me greatly in providing a better understanding of GCT.

I thank Rachel Ingrassia and Deborah Kloppenburg of Electrical and Computer Engineering for their help with all the paperwork. Erin Keffeler, my International Adviser has been very helpful throughout my stay at Stony Brook.

I consider myself lucky to have such good friends; they have constantly inspired me, shown me love, joyfulness and wisdom. It is impossible for me to name everyone here who have left an impression on me and have made me who I am. I must name a few who have been particularly demanded more from me. I owe my thanks and much more to my dear friends Arjun, Karthik and Gopal for guiding me and for nurturing my dreams. I thank all my friends in Stony Brook for making my stay both memorable and enjoyable.

Lastly, I thank my family for all their sacrifices. I am indebted to my mother Vishalaskshi and my father Shivashankar; their love and support have kept me going in tough times. I thank my sister Shubha and my brother in-law Kiran; without their support this work would not have been possible.

I am solely responsible for any remaining errors in this thesis. If any errors are found, they will be corrected through posting of “Corrections” or “Errata” for this thesis on our lab website.

1. Introduction

In the past decade, there has been an exponential growth in the computing power of microprocessor chips. This combined with significant reduction in the cost of digital cameras have increased the amount of image data captured and processed on the computer. The advantage of digitizing an image is the access to several algorithms or image processing techniques that enhance the quality of the image. The term quality here and throughout the work refers to the perception of image quality to a human observer. A new field called Computational Photography is emerging that challenges the traditional boundaries of photography [1]. New kinds of imaging devices are being developed that obtain a customized image (of a scene) after processing the raw image using a computer [2, 3]. In general, sophisticated algorithms are making novel and complex manipulations of image data possible with imaging devices.

One such topic where computer processing is useful is restoration of images degraded by blur. This work focuses on algorithms that restore 2D and 3D blurred images (due to defocus) in a computationally efficient manner. Computational efficiency is achieved by using new models of image formation based on the physical properties of typical imaging systems. For example in 2D, a concept called *localization*, which asserts that blurring is a local phenomenon is used to reduce computational

complexity of image deblurring. In 3D image restoration, computational efficiency is achieved through deblurring in a new domain. These approaches also provide some new insight and perspectives to image restoration.

1.1. Motivation

Imaging devices are used for a wide variety of applications from Machine Vision, Astronomy, 3D Microscopy, Medical Imaging etc. The following paragraphs present a brief summary of some applications of image deblurring algorithms.

A typical machine vision task is the inspection of a manufactured product for quality (not to be confused with image quality). An example machine vision application is: Rejecting pencils that do not have erasers properly attached at the end. The task is done in the production line itself so as to avoid error propagating from one stage of production to the other. The challenge is to perform the inspection task quickly so that production is not slowed down. The imaging devices used in Machine Vision applications suffer from blurring caused by defocus or motion. The information in an image is contained in the edges or boundaries of an object. Blurring smoothes the edges and hence some information is lost due to degradations. In such cases, image deblurring algorithms have to be used to process the images before performing an inspection task.

Microscopes are used to magnify and record images of very small objects. In Fluorescence Microscopy, different parts of a cell, e.g. cell nucleus, are stained by a dye that is illuminated by specific wavelengths of light. Several images of the illuminated set of points are taken by moving the specimen along the axis perpendicular to the imaging device. The observed set of images is a 3D blurred image, which is deblurred

by suitable algorithm to recover the 3D volumetric distribution of light [4]. It is physically impossible to record a 3D structure without defocus. Hence the 3D blurred image is restored. It then reveals the 3D structure of the object (the cell nucleus). Such 3D structure helps us learn about the bio-chemical processes at the microscopic level. In medical imaging analysis, deblurring may be used to enhance the quality of the image to assist a physician.

Other applications include deblurring telescopic images. For instance, deblurring Hubble Space Telescope (HST) images is well documented in literature[5, 6]. Recently, cameras with adaptive optics have been developed that correct themselves based on the degradations present in the observed data[7].

Deblurring algorithms provide a way for overcoming imperfections of camera optics. With an accurate model of the imaging device, the observed data is deblurred such that the degradations due to camera optics are minimized. The algorithms also offer many insights useful for modeling other imaging modalities such as Radar imaging, Ultrasound imaging etc. The principles of image deblurring are also used in medical image reconstruction and vice versa.

1.2. Why does blur occur?

Before understanding the deblurring methodologies, some fundamentals of imaging itself have to be discussed. The simplest imaging device is an opaque planar object with a *pin-hole* in it such that light from one side of the plane travels through the hole to the other side. Such a device is called a pin-hole aperture camera or a pin-hole camera. The scene, from which light travels to the pin-hole, is said to be in front of the camera. Hence, the image of the object is formed behind the camera. It is possible to record a

real (inverted) and nearly sharp image of the whole scene by placing a detector parallel to the pin-hole plane. That is, every point in the image obtained by pin-hole camera is focused irrespective of the location of the scene points. However, due to the small size of the aperture the amount of light passing through it is very small. The recorded images would be degraded by noise, or would require a long time of exposure to collect sufficient amount of light. Thus the pin-hole camera is impractical.

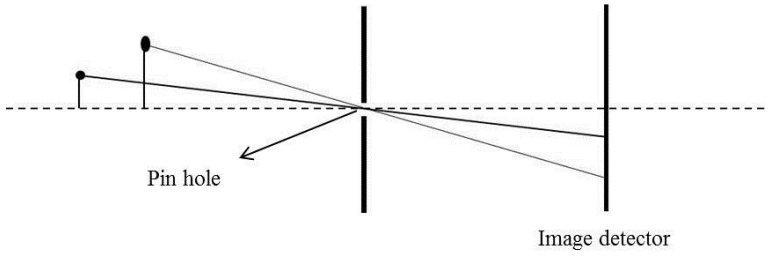


Figure 1.1: Two point objects being imaged by pin-hole camera. The camera forms a real and focused image everywhere along the detector irrespective of the location of the objects from the camera.

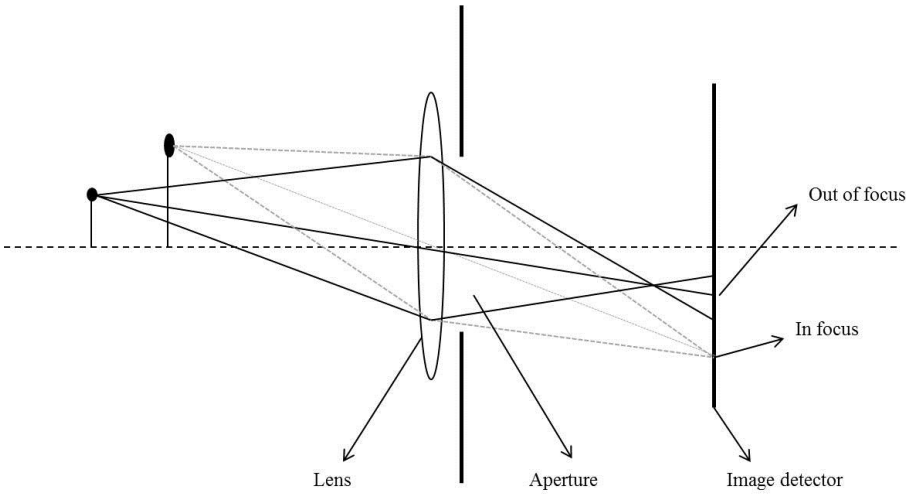


Figure 1.2: Two point objects imaged by a finite aperture thin lens camera. In the figure the detector is placed such that the point object closer to the lens is in focus. The farther point object is out of focus. The position of the focused image for a given object depends on the location of the object and the focal length of the lens.

Lenses are used to bend wavefronts such that they converge to form a point image. Also, lenses collect more light to produce better quality images. However, the aperture is quite large compared to a pin-hole. This causes only the points located at particular distance from the lens come to focus at the image detector; as governed by the lens maker's formula. The points that do not come to focus are said to be defocused or blurred. A point object that is defocused could be brought to focus by moving the image detector along the optical axis (shown as dashed line in Fig. 1.2). Thus the amount of defocus thus depends of the detector location and the location of the object.

1.2.1. Shift-variant degradation

Since points at different distances from the lens come to focus at different image detector locations, an image of a 3D object suffers different amounts of degradations at

different points in the image. That is, the amount of blur varies from pixel to pixel. Hence, defocus blur is said to be shift-variant. The shift-variant characteristic is also observed in image degradations caused by lens aberrations and motion blur. The simplest motion blur example is that of an image taken from a moving car. The trees that are close appear smeared or blurred whereas objects that are farther, like mountain landscape, suffer relatively less degradation. This is due to the fact that the amount of magnification of trees is greater than that of mountain landscape. Lens aberrations occur noticeably due to imperfections of the camera optics.

1.3. Image deblurring process

Shift-variant deblurring in two and three dimensions is the focus of this work. The principle behind image deblurring is as follows. The image forming system is modeled mathematically such that the forward operation of the model produces a blurred image as the output of the camera system with focused image as the input, as shown in Fig. 1.3.

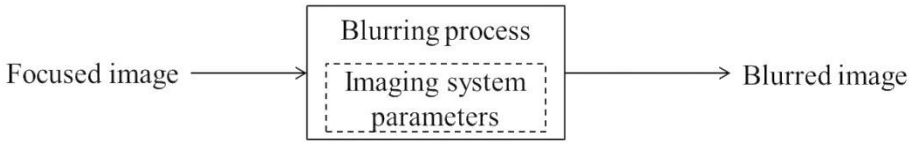


Figure 1.3: Block diagram of a mathematical model of an imaging system. Focused image is the input and blurred image is the output. The system is said to perform a *forward* operation.

The deblurring problem is schematically represented by Fig. 1.4. Deblurring algorithms discussed in this report restore blurred images recorded by a system whose mathematical model is known. Obtaining the model parameters for given blurred

images is a research topic, generally referred to as image identification or blur identification [8-13]. There exist some methods called blind deconvolution algorithms that treat both, the model parameters and the focused image as unknowns [14, 15].

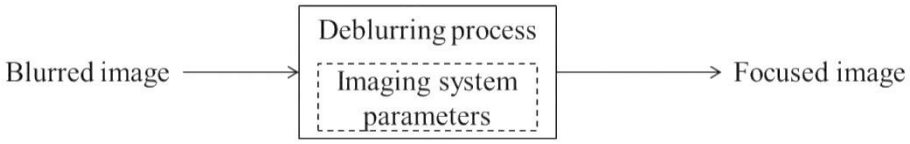


Figure 1.4: Showing conceptual diagram of deblurring process. The deblurring algorithm accepts blurred image as the input and produces the focused image as the output for a given imaging system. Deblurring is therefore the opposite of Fig. 1.3, hence is said to perform an *inverse* operation.

Since deblurring is an inverse problem, it is often numerically ill-conditioned or may not have a unique solution; i.e., it may be ill-posed[16, 17]. It is also computationally intensive to solve a deblurring problem, generally due to the large size of image data. A good restoration algorithm must:

- be computationally efficient
- be accurate in the mean squared error sense
- be able to handle noisy input (blurred image)
- not introduce visible blocking artifacts

Artifacts here mean artificial patterns introduced at the time of processing. There are tradeoffs between some requirements listed above. For example, there is a tradeoff between accuracy and computational efficiency. The goal of this work is to device algorithms that sufficiently meet all the above requirements.

1.4. Report organization

The rest of the report is organized as follows. In chapter 2, fundamentals of image formation, mathematical models of imaging systems and the prior art in shift-variant image restoration are discussed.

In Chapter 3, a new deblurring algorithm that effectively recovers focused image from highly defocused images is investigated. It is an iterative restoration algorithm. The algorithm is tested on simulation experiments and real data.

Chapter 4 presents a localized and computationally efficient iterative approach to shift-variant image restoration. Two algorithms are investigated and compared with other techniques for accuracy, computation time, robustness against noise and quality of restored images. The algorithms also offer new insight to ill-conditioned nature of deblurring problems.

Chapter 5 introduces a new theorem called the Generalized Convolution Theorem (GCT). GCT is used to transform a shift-variant system to a shift-invariant system through domain transformation. Proof of the theorem is presented. The conditions under which the transformation is possible are also highlighted. The theorem is verified in 1D.

Chapter 6 investigates two applications of GCT. First is 3D deconvolution microscopy. A shift-variant 3D imaging system is transformed to a shift-invariant system using GCT. A blurred 3D image is restored in the transformed domain in a computationally efficient manner. Through simulation experiments it is demonstrated that GCT improves both accuracy and computational efficiency of shift-variant image restoration. Next, 3D shape recovery using GCT is briefly investigated.

Chapter 7 summarizes the report and provides some directions for future work.

2. Background and Related Work

The approach to shift-variant image restoration was introduced in the previous chapter at a high level. The steps involved are discussed in more detail here. Image formation is modeled mathematically as a linear shift-variant system. Then, several shift-variant image restoration algorithms are presented. The purpose of this chapter is to make the reader familiar with the fundamentals of image formation and standard techniques for processing shift-variant blurred images to recover a focused image estimate.

The rest of the chapter is organized as follows. The theory of image formation under *paraxial optics* assumption is presented in Section 2.1. In Section 2.2, the linear shift-variant (LSV) model for image formation is developed. Section 2.3 further discusses the impulse response of LSV model called the *point spread function* (PSF). Various shift-variant image restoration techniques are presented in Section 2.4.

2.1. Image formation in a thin lens

Camera lenses form real images of the object that are recordable (observable) on a detector. Present day cameras have a set of lenses that guide the light rays to form high-

quality images. However, for the purposes of the work reported here, it is sufficient to assume that a camera is a single thin lens; see Fig. 2.1. Also, it is assumed that the aperture (D in Fig. 2.1) is small enough so that the light rays from the object travel almost parallel to the optical axis. This is the *paraxial optics* assumption. The figure below shows a schematic diagram of thin lens image formation. Note that this is a cross section and the rays in the figure are all in the YZ plane that contains the optical axis.

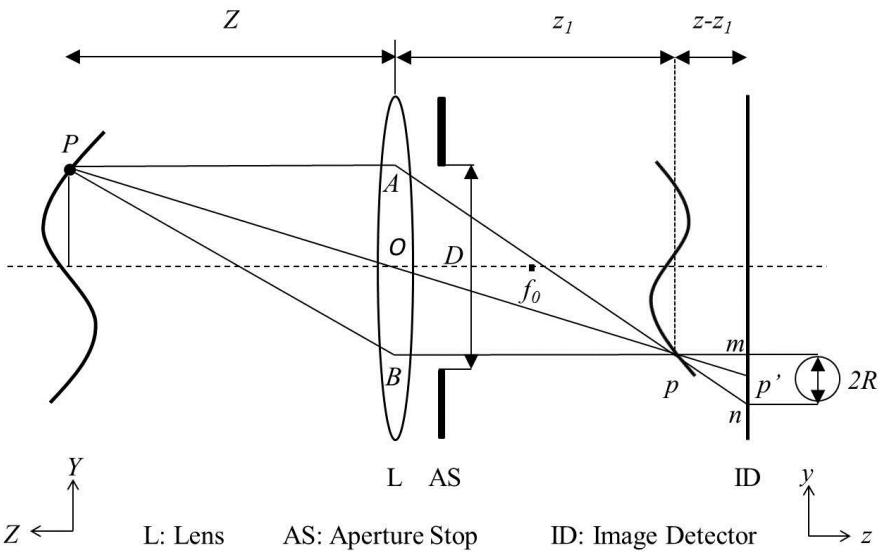


Figure 2.1: Schematic diagram of image formation of a curved surface (a cross section through the axis). The world coordinates are to the left of the lens and image coordinates are to the right. The curve in image coordinates is the set of Gaussian image points of the real surface.

A point P on the surface located at a distance Z from the lens center O, comes to focus at a distance z_1 from the lens center. The light ray passing from P to O ($\overrightarrow{PO}$) remains undeflected; this is called as the *chief ray*. Rays $\overrightarrow{PA}$ and $\overrightarrow{PB}$ undergo refraction and meet the chief ray behind the lens at location p. These rays are known as *marginal rays*. The point p at which the rays converge is called the Gaussian image point. A focused image is said to be formed on a plane perpendicular to the optical axis, if the

image detector plane contains the Gaussian image point. Under this condition, a thin lens satisfies the following equation:

$$\frac{1}{f_0} = \frac{1}{Z} + \frac{1}{z_1} \tag{2.1}$$

In Eq. (2.1), f_0 is the *focal length* of the lens, and Z and z_1 are the *object distance* and the *image distance* respectively. However, when the image detector is translated in a direction parallel to the optical axis away from the Gaussian image point, the image of the point object is said to be out-of-focus or *defocused*. As shown in the figure, the image of the point object is then a circular patch, known as *blur circle* and its radius is called the *blur circle radius*.

A surface, when being imaged by the lens, forms a surface of focused image points behind the lens, as shown in the figure. However, due to image detector being planar, this surface is not directly recordable in a camera. Often, the images recorded have different amounts of blur in different regions (based on the image detector position for the object). Hence, observed images are in general shift-variant blurred. From this, it is apparent that only a planar object perpendicular to the optical axis can produce a perfectly focused image.

2.2. Mathematical modeling of image formation

Image formation is modeled by a linear shift-variant system[18-20]. It is given by the *superposition integral* (also known as the Fredholm integral equation of the first kind).

$$g(x,y) = \int_a^b \int_c^d h(x,y,u,v) f(u,v) du\, dv\,. \tag{2.2}$$

In Eq. (2.2), $g(x,y)$ is the observed (or blurred) image, $f(u,v)$ is the focused image and $h(x,y,u,v)$ is the shift-variant impulse response of the linear system. For a point object, when it is in focus, the impulse response is given by the Dirac delta function. However, when the object is defocused, the point image gets spread out hence it's known by the name *point spread function* (PSF). The function $h(x,y,u,v)$ is interpreted as a set of 2D PSFs, where each PSF corresponds to a different pixel. Due to the assumption that light energy is conserved by the optical system, the PSFs (at every location) satisfy the following condition:

$$\int h(u,v)\, du\, dv = 1 \tag{2.3}$$

Defocus is just one type of aberration of an optical system. For a non-paraxial imaging system, *primary aberrations* Spherical, Coma, Astigmatism, Field Curvature and Distortion become significant. The images degraded by primary aberrations are also described by Eq. (2.2). Thus, different kinds of degradations are described by different PSF models. The focus of this work is mainly towards restoring images degraded by defocus. However, for completeness all the PSF models are introduced in the following section. It is noted that shift-variant image restoration in a general sense includes other aberrations.

2.3. PSF models

In this section, analytical expressions for defocus PSF models are presented. The expressions are derived from *geometric optics*. In Chapter 6, a brief derivation of geometric optics PSF is given. The aberration PSFs are obtained through *wave optics*. The theory of primary aberrations is quite involved and is therefore skipped here. Instead, some images are given that highlight the shape of PSFs of different *primary aberrations*.

2.3.1. Defocus PSF models

In geometric optics, the defocus PSF for a point object at (x, y) is centered at the same location [21]. Two models for the PSF are frequently used. One of them is the cylindrical PSF model given by,

$$h(x, y, u, v) = \begin{cases} \dfrac{1}{\pi R^2(x, y)} & \text{if } (x-u)^2 + (y-v)^2 < R^2, \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

In Eq. (2.4), R is the *blur circle radius*. It is a function of the Gaussian image point location and the image detector position. The *blur circle radius* and in turn the PSF is parameterized by camera constants like *focal length* f_0 and *aperture diameter* D. The second model is the Gaussian PSF model given by

$$h(x, y, u, v) = \frac{1}{2\pi\, \sigma^2(x, y)} \exp\left(\frac{(x-u)^2 + (y-v)^2}{\sigma^2(x, y)}\right). \tag{2.5}$$

The standard deviation $\sigma(x, y)$ determines the width of the PSF. It can be shown that σ is proportional to the *blur circle radius* and the following equation provides a very good approximation to the relation.

$$R = \sqrt{2}\, \sigma \tag{2.6}$$

The figure below shows mesh – grid visualization of Cylindrical and Gaussian PSFs.

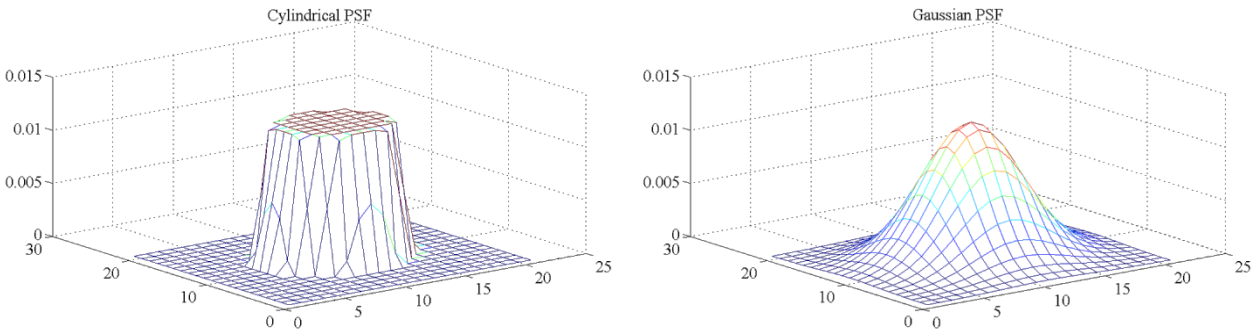
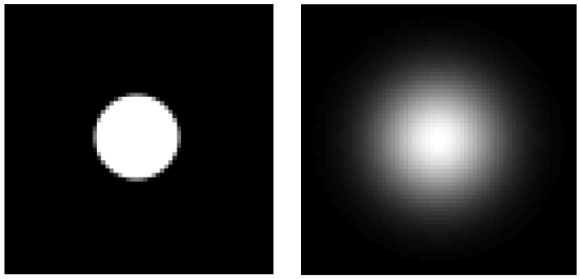


Figure 2.2: Showing Cylindrical PSF of radius $R = 5$ pixels and Gaussian PSF of $\sigma = {}^{5}/_{\sqrt{2}}$ pixels. The mesh-grid plots were generated on 21×21 array.

Note that from Figures 2.2 and 2.3, the assumption of local support domain of the geometric optics PSF models is valid. The Cylindrical PSF is zero outside the *blur circle*. Although the Gaussian PSF theoretically has infinite support, the values decay exponentially. Therefore, the PSF can be truncated beyond twice the standard deviation from the mean.



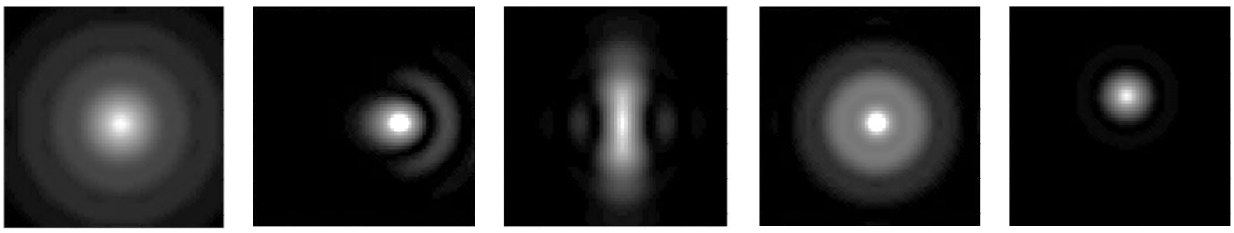
(a) (b)

Figure 2.3: Showing images of (a) Cylindrical and (b) Gaussian PSF model

2.3.2. Primary aberration PSFs

Thin lens imaging systems are affected by aberrations when the paraxial optics assumption is not valid. The *primary aberrations* except Spherical aberration are shift-

variant even for a planar perpendicular object. The amount of aberration depends on the axial distance or depth of the object point. The PSFs are obtained by squaring the Fourier Transform of the pupil function of the imaging system. A pupil function is a complex exponential describing the phase of the wavefront (of light from the point object) at different image-coordinate points. The reader is suggested some references for further reading [19, 22, 23]. The figure below shows examples of all the *primary aberrations*.



(a) (b) (c) (d) (e)

Figure 2.4: Sample primary aberration PSFs. (a) Spherical, (b) Coma, (c) Astigmatism, (d) Field Curvature and (e) Distortion (tilt).

2.4. Shift-variant image restoration

In the previous section, a mathematical model for thin lens imaging system was developed that relates a *blurred image*, the system's *point spread function* and the underlying *focused image*. The problem of shift-variant image deblurring amounts to solving the *superposition integral*. Eq. (2.2) can be written in a matrix form $\mathbf{g} = \mathbf{Hf}$. Hence, solving for an unknown focused image with a given blurred image and a known PSF is equivalent to solving a linear system of equations. In other words, a correct solution to the focused image estimate satisfies the linear system of equations. An image of size $N = m \times n$, where m is the height and n is the width, produces N equations

and the same number of unknowns. Although straightforward solutions are possible with such problem formulation, they are highly computationally inefficient. Moreover, there is an added problem of the linear systems being ill-conditioned which means that the system is highly sensitive to perturbations (noise) in the input, or ill-posed which means a unique solution may not exist.

A shift-invariant blurring operation is also a linear system of equations similar to the one described above. The key difference is that *shift-invariant* blurring is described by the *convolution integral*. The convolution integral is a special case of the superposition integral and has been extensively studied in signal processing[24, 25]. Due to a special property of the Discrete Fourier Transform (DFT), the convolution integral is expressed as a product in the frequency domain. Therefore, shift-invariant deblurring, also known as Deconvolution, is efficiently implemented using the Fast Fourier Transform (FFT) [26-31]. However, it is not possible to directly apply FFT for a shift-variant deblurring problem.

Ill-posedness of the deblurring problem is in general handled by regularization technique, which adds additional constraints that a solution must satisfy. Shift-variant deblurring is more challenging than Deconvolution as it is computationally intensive to implement a straightforward solution. Therefore, the challenge is to develop algorithms that are computationally efficient and yet accurate. In the next section, a survey of the different approaches to shift-variant image restoration is presented.

2.4.1. Literature survey

Earlier, it was said that shift-variant deblurring is equivalent to solving a large linear system. Since direct inversion of such systems is not practical, iterative techniques are

employed to solve the problem. One such method was provided by Landweber[32]. Landweber's method finds the best estimate for focused image that minimizes the residual (gradient) between the actual blurred image and a blurred image estimate in an iterative fashion. This method has been applied in image reconstruction as well as image deblurring [33] to restore shift-invariant images. An iteration of the algorithm requires computing one *blurring operation* which is calculating the blurred image estimate from the solution. Landweber's method is known to provide accurate solutions. However, its convergence is linear and hence slow. Conjugate gradient methods improve the convergence of the gradient based approach [34-37]. Another iterative technique is the Lucy-Richardson algorithm [38, 39]. Lucy-Richardson algorithm multiplicatively updates the focused image estimate. It incorporates non-negativity constraint within the algorithm, which is necessary to obtain good quality restoration. However, unlike an additive updating scheme, Lucy-Richardson algorithm requires two blurring operations per iteration.

The above iterative schemes are computationally quite expensive. An assumption that the variation in blur is small within small regions, say 32×32, is made that enables reduction in the computational cost of shift-variant deblurring. A shift-variant blurred image is divided into multiple shift-invariant blocks, within which blurring operation is expressed by convolution [8, 40]. This is the principle behind sectioning methods investigated by Trussel and Hunt [13, 41-43]. Nagy et al have investigated an iterative conjugate gradient method based on the same principle [35, 36, 44].

An alternative to iterative techniques is offered by direct methods. They express convolution in each block as a matrix-vector product. The resulting system matrices are either Block Toeplitz or Block Circulant, which are solved using singular value decomposition (SVD). However, to decompose the kernel efficiently, it is also required

to assume that the kernel (or the PSF) is separable. Kamm et al have investigated SVD based methods that invert the linear system efficiently using the Kronecker product [27, 45-48]. The sectioning shift-variant methods are computationally efficient, however, are prone to blocking artifacts that can reduce the quality of the focused image estimate. The sectioning methods also tend to be less accurate than their exact iterative counterparts. Another way of reducing computational cost is by using coordinate transformations to warp a shift-variant blurred image such that shift-invariance holds true in the new space[49, 50]. However, this has been possible for only motion blurring and Coma in two dimensions. Time-frequency approaches though theoretically interesting are quite computationally expensive [51, 52].

Since blurring is a local phenomenon and this observation was used to localize the blurring and deblurring operations, an S-Transform based sectioned method was investigated for shift-variant deblurring [8]. Later, a localization transformation called Rao Transform (RT) was introduced by Subbarao [53-56]. A region comparable to the size of the support domain of the PSF is expressed in terms of truncated Taylor's series. This is done for each pixel. The blur integral is then transformed to a partial differential equation that expresses a blurred pixel value as the weighted sum of focused image pixel and its spatial derivatives at that location. The weights depend on the radius of the blur circle. The single-interval RT (SRT) method derives a small linear system from the partial differential equation by differentiating it further. The solution to this local linear system provides an estimate of the focused image. Such localized methods do not produce blocking artifacts and provide reasonably accurate solutions [53].

In the following sections some of the algorithms in prior art that are used for comparison are discussed in more detail.

2.4.2. Gradient based iterative approach

One of the earliest methods to solve the *superposition integral* was the gradient based iterative approach [32]. It is given as follows.

$$\hat{f}_{k+1}(x,y) = \hat{f}_k(x,y) + \alpha\left[g(x,y) - \int_a^b \int_c^d h(x,y,u,v)\hat{f}_k(u,v)du\,dv\right]. \quad (2.7)$$

The term inside the square braces is the *gradient* in the L_2 space. The above equation is also written in a matrix form that is the standard gradient descent approach to solving linear systems. The iterative method does not make any assumptions or approximations about the kernel, therefore it is exact. However, it has been shown that it has linear convergence. The convergence is also seen to depend on the size of the support domain of the kernel [35]. The step size parameter α is often modified to improve convergence [36]. One such method is found in [33]. There the gradient at k^{th} iteration is denoted by $\nabla \mathrm{f}_k$. The step-size parameter is expressed in terms of the gradient as follows.

$$\alpha_k = \frac{\|\nabla \mathrm{f}_k\|_2}{\|\nabla \mathrm{f}_{k-1}\|_2} \quad (2.8)$$

For the first iteration the mean squared value of the gradient $\|\nabla \mathrm{f}_1\|_2$ could be used. The value of α_k however has to be truncated at $2/\sigma_1^2$; σ_1 is the largest singular value of the kernel matrix (for details see [33]). The computational complexity of Eq. (2.7) is $O(k^2N)$ for a shift-variant kernel having $k \times k$ support domain. The possibility of reducing the complexity of Eq. (2.7) through localization is investigated in Chapter 4.

Other gradient based methods include the Conjugate-gradient family of methods. Nagy et al have investigated iterative Conjugate-gradient methods for fast convergence [35, 36, 57]. However, the assumption of blockwise shift-invariance is made to improve

the computation time of iterations. Youn-sik has compared the performance of the SRT method with the Conjugate-gradient methods [57]. The Projection on to Convex Sets method is related to gradient based iterative methods [58].

2.4.3. The Lucy-Richardson algorithm

Gradient based methods improve the estimate additively while a multiplicative update equation has come to be known by the name Lucy-Richardson algorithm after its two (independent) inventors [38, 39]. The Lucy-Richardson update equation for shift-variant image restoration is given below:

$$\hat{f}_{k+1}(x,y) = \hat{f}_k(x,y) \int_a^b \int_c^d \frac{g(u,v)}{\hat{g}_k(u,v)} h(x,y,u,v)\, du\, dv \ . \tag{2.9}$$

In Eq. (2.9), $\hat{g}_k(u,v)$ is given by:

$$\hat{g}_k(u,v) = \int_a^b \int_c^d \hat{f}_k(\alpha,\beta)\, h(u,v,\alpha,\beta) d\alpha\, d\beta \ . \tag{2.10}$$

Thus, the iterative method has two *blurring* (forward) operations per iteration. The method has a Bayesian interpretation that has been used in non-linear MAP statistical image restoration algorithms [59, 60]. The method has an inbuilt positivity constraint that is very useful in image restoration. In this work, Lucy-Richardson algorithm is used to restore 3D shift-variant blurred images; Chapter 6.

2.4.4. Image restoration by spectral filtering

The direct methods, which estimate focused image in one step by inverting the imaging system, are discussed here in detail. Consider an LSV system in 1D as follows:

$$g(s) = \int_a^b h(s,t) f(t)\, dt \tag{2.11}$$

The above equation is discretized by numerical quadrature [61, 62] and written in the form of a matrix-vector equation below.

$$\mathbf{g} = \mathbf{H}\mathbf{f} \tag{2.12}$$

In Eq. (2.12), let the length of vectors $\mathbf{g}$ and $\mathbf{f}$ be N. Each component of the vectors represents a sub-interval of interval $[a, b]$. Naturally, the matrix $\mathbf{H}$ is of size $N \times N$. The simplest way to obtain a solution to $\mathbf{f}$ in Eq. (2.12) is to invert the matrix $\mathbf{H}$. The generalized inverse or pseudo inverse of $\mathbf{H}$ may be used to obtain a solution. For a 2D image, a system of the form in Eq. (2.12) can be arrived at by lexicographically ordering pixels as the components of vectors $\mathbf{g}$ and $\mathbf{f}$.

The above approach has mainly two properties. The system matrices obtained with the above approach are often ill-conditioned [17, 61, 62]. That is, they highly are sensitive to perturbations during inversion. Sometimes the problem may be ill-posed, in that the solution may not be unique. In such cases some form of *regularization*, which is an elegant technique to produce meaningful solution in the presence of noise, has to be incorporated [16, 63] . The least-squares minimization, given below, is often used with positivity constraint to achieve meaningful results.

$$\min_{\mathbf{f}} \quad \left\{ \|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2 \right\} \tag{2.13}$$

The pseudoinverse $(\mathbf{H}^\mathbf{T}\mathbf{H})^{-1}\,\mathbf{H}^\mathbf{T}$ is a closed form solution to Eq. (2.13).

The second property is related to the computational complexity of evaluating Eq. (2.12) or (2.13). The complexity of evaluating the two equations grows rapidly with the size of the image. For example, a small image of size 100×100 pixels, the size of the system matrix that has to be inverted is of size $10^4 \times 10^4$. Therefore, in practice, this

approach is never used. Instead, an image is divided into small shift-invariant blocks of say 32×32 or 64×64 and each block is inverted independently. The shift-invariant blocks (or convolution approximations) are expressed by a Toeplitz or a Circulant system matrix, depending on the boundary conditions. The following provides a general overview of matrix-based methods for image restoration.

The matrix **H** is decomposed as follows:

$$\mathbf{H} = \mathbf{F}^{*}\mathbf{\Lambda}\,\mathbf{F} \tag{2.14}$$

In Eq. (2.14), **F** is the Fourier basis and **Λ** contains the coefficients of spectral components of **H**. Then, by filtering the spectral components, noise amplification is minimized thereby achieving regularization. Such methods are analyzed using singular value decomposition as follows:

$$\mathbf{H} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V^{T}}$$

$$\Rightarrow \hat{\mathbf{f}} = \sum_{i=1}^{N} \frac{\mathbf{u}_i^T\,\mathbf{b}}{\sigma_i}\,\mathbf{v_i} \tag{2.15}$$

By scaling the singular values σ_i with filter factors φ_i, regularization is achieved. Thus the focused image estimate $\hat{\mathbf{f}}_{reg}$ is obtained:

$$\hat{\mathbf{f}}_{reg} = \sum_{i=1}^{N} \varphi_i\,\frac{\mathbf{u}_i^T\,\mathbf{b}}{\sigma_i}\,\mathbf{v_i} \tag{2.16}$$

The simplest form of regularization is the Truncated SVD (TSVD), where the filter factors cutoff singular values after a specified number. That is,

$$\varphi_i = \begin{cases} 1, & i = 1, \dots, k, \\ 0, & i = k+1, \dots, N. \end{cases} \tag{2.17}$$

The Tikhonov regularization method is obtained if the filter factors are defined as follows:

$$\varphi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}, \quad i = 1, \ldots, N \tag{2.18}$$

The Tikhonov regularization method solves the following optimization problem.

$$\min_{\mathbf{f}} \left\{ \|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2 + \alpha^2 \|\mathbf{f}\|_2^2 \right\} \tag{2.19}$$

The generalized Tikhonov regularization method solves the following minimization problem.

$$\min_{\mathbf{f}} \left\{ \|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2 + \alpha^2 \|\mathbf{D}\mathbf{f}\|_2^2 \right\} \tag{2.20}$$

In the equation above, $\mathbf{D}$ is the Laplacian operator. Eq. (2.20) has the following closed form solution, derived by Hansen et al [63].

$$\hat{\mathbf{f}}_{reg} = (\mathbf{H}^\mathbf{T}\mathbf{H} + \alpha^2 \mathbf{D}^\mathbf{T}\mathbf{D})^{-1}\mathbf{H}^\mathbf{T}\mathbf{g} \tag{2.21}$$

The above closed form solutions are not suitable for implementation. Instead they suggest a quadratic minimization of Eq. (2.20) using Newton's method. Although the methods are analyzed using SVD, the implementation uses Fast Fourier Transform (FFT) to perform restoration using Shift-invariance approximation, thereby reducing computational requirements. As a consequence, the accuracy of the solution is reduced. The restored images contain blocking artifacts that introduce discontinuity at the block boundaries. Therefore, quality of the restored image is compromised.

2.4.5. Localized shift-variant image restoration

Section 2.4.4 provided a brief overview of matrix-based methods for shift-variant image restoration. They are global in nature, in the sense that the focused image estimate is recovered through inversion of large (global kernel) matrices. However, the blur kernel due to defocus has limited support domain that is often small compared to the size of the image. For example, a PSF may be of size 11×11 whereas, images are often much larger than that. This local support is exploited to reduce the computation complexity of the superposition integral. This is achieved through a localization transformation called the Rao Transform (RT)[53]. The RT method for shift-variant image restoration is a matrix-based method called single-interval RT (SRT) method. The details of the SRT method are presented here.

A blurred pixel g at (x, y) receives contribution from any pixel whose blur circle encloses the current location (x, y). Using change of variables the LSV system can be made to express blurred pixel as weighted sum of focused image pixels from a local neighborhood centered at the current pixel. This is the central idea of RT and is illustrated below. Shift-variant imaging is expressed by the following expression.

$$g(x,y) = \int_b^a \int_d^c k(x,y,u,v)\, f(u,v)\, du\, dv \tag{2.22}$$

Substituting $u' = x - u$ and $v' = y - v$ in Eq. (2.22), the following is obtained:

$$g(x,y) = \int_{x-b}^{x-a} \int_{y-d}^{y-c} k(x-u, y-v, x, y)\, f(x-u, y-v)\, du\, dv \tag{2.23}$$

By defining a new localized kernel $h(x, y, u, v) \equiv k(x + u, y + v, x, y)$, the above equation is rewritten in the following localized form.

$$g(x,y) = \int_{x-b}^{x-a} \int_{y-d}^{y-c} h\,(x-u, y-v, u, v\,)\, f(x-u, y-v)\, du\, dv \tag{2.24}$$

The m^{th} partial derivative with respect to x and n^{th} partial derivative with respect to y of a function f of variables x and y is denoted by $f^{(m,n)}$. A similar convention is followed for other functions. The p^{th} moment with respect to u and q^{th} moment with respect to v of h is denoted by $h_{p,q}$. Then the various moments of the PSF and its derivatives are defined by:

$$h_{p,q}^{(m,n)} = \int_{x-b}^{x-a} \int_{y-d}^{y-c} u^p\, v^q\, h^{(m,n)}(x,y,u,v) du\, dv\,. \tag{2.25}$$

In the equation above, $m, n, p, q = 0, 1, 2, \ldots$. The truncated Taylor's series expansion of $h(x-u, y-v, u, v)$, with respect to (x, y, u, v) is given below.

$$h(x-u, y-v, u, v) = \sum_{m=0}^{M} a_m \sum_{j=0}^{m} C_j^m\, u^{m-j}\, v^j\; h^{(m-j,j)} \tag{2.26}$$

In Eq. (2.26) $a_m = \frac{(-1)^m}{m!}$ and C_j^m is the binomial coefficient of the Taylor's series. Similarly, the truncated Taylor's series expansion of $f(x-u, y-v)$ about (x, y) is given by:

$$f(x-u, y-v) = \sum_{n=0}^{N} a_n \sum_{j=0}^{m} C_i^n\, u^{n-i}\, v^i\; f^{(n-i,i)} \tag{2.27}$$

Substituting Eqns. (2.26) and (2.27) in Eq. (2.24) and changing the order of summations and integration, the following expression is obtained.

$$g(x,y) \approx \sum_{n=0}^{N} a_n \sum_{j=0}^{m} C_i^n\; f^{(n-i,i)} \sum_{m=0}^{M} a_m \sum_{j=0}^{m} C_j^m\; h_{m+n-i-j,i+j}^{(m-j,j)} \tag{2.28}$$

The coefficients of $f^{(n-i,i)}$ above are written in a concise form below,

$$S_{n,i} = a_n C_i^n \sum_{m=0}^{M} a_m \sum_{j=0}^{m} C_j^m \; h_{m+n-i-j,i+j}^{(m-j,j)} . \tag{2.29}$$

By substituting Eq. (2.29) in Eq. (2.28), the expression for *forward* RT is obtained:

$$g(x,y) \approx \sum_{n=0}^{N} \sum_{j=0}^{m} S_{n,i} \, f^{(n-i,i)} \tag{2.30}$$

A linear system is obtained by differentiating the above expression w.r.t variables x and y up to order N and truncating the derivative terms of order higher than N.

$$\begin{bmatrix} g^{(0,0)} \\ g^{(1,0)} \\ \vdots \\ g^{(0,N)} \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & \cdots & \cdots \\ r_{10} & r_{11} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ \cdots & \cdots & & \cdots \end{bmatrix} \begin{bmatrix} f^{(0,0)} \\ f^{(1,0)} \\ \vdots \\ f^{(0,N)} \end{bmatrix}$$

$$\Rightarrow \mathbf{g_{x,y}} = \mathbf{R_{x,y}} \, \mathbf{f_{x,y}} \tag{2.31}$$

Eq. (2.31) is the localized linear system obtained through RT. The above expression is localized in the sense that at each pixel, the blurred image value is expressed in terms of the local information (derivatives) of the focused image. The above system of equations contains $(N+1)(N+2)/2$ unknowns and equations. The solution to the focused image $f^{(0,0)}$ is obtained by solving it with standard techniques like TSVD or Tikhonov regularization.

The SRT method is highly computationally efficient and is also parallel. It is possible to obtain a closed form solution to focused image based for defocus PSF models derived from geometric optics. The SRT system matrices are typically of size 7×7 or 10×10 and therefore solving the system is not as expensive as SVD based methods. Moreover, since no sectioning is made, no discontinuity or blocking artifacts are produced in restored images. However, the truncated Taylor's series expansion hinders the

performance of SRT method for large amounts of defocus. This problem is addressed through the multi-interval RT (MRT) method in Chapter 3.

In spite of SRT being computationally efficient, its performance could be improved by solving the system iteratively rather than through a direct inversion. Iterative implementation provides more control over the progression of the solution towards a focused image estimate. This approach is investigated in Chapter 4.

2.5. Summary

In this chapter, image formation in a thin lens was presented. The optical system was modeled as a linear shift-variant system. The relation between a blurred image, its underlying focused image and the optical system was expressed as a superposition integral. The problem of deblurring image was stated as equivalent to solving the superposition integral. The Landweber's method and the Lucy-Richardson algorithm were presented in detail. Computationally efficient sectioning methods like SVD based Tikhonov regularization was also discussed in detail. Finally, localized approach to shift-variant image deblurring (SRT) was presented in detail.

The sectioning methods presented in this chapter produce blocking artifacts that can reduce the quality and accuracy of the focused image estimate. The exact iterative methods are very expensive to implement. Although the SRT method avoids blocking artifacts at lower computational cost, it has some drawbacks. Mainly, the SRT methods perform well for images blurred by low amount of defocus. Moreover, inverting a small linear system at every pixel is not necessarily the best approach to finding the focused image estimate. In Chapter 3, the MRT method is investigated, which alleviates the issue of restoring non smooth (high-frequency) images. Then in Chapter 4, the SRT

method is modified to produce an iterative update equation that makes the method more computationally efficient and stable.

3. Restoring Highly Defocused Images

In the previous chapter the fundamentals of shift-variant image restoration were presented. RT based matrix method called single-interval RT (SRT) was also discussed in detail. The computationally efficient SRT method exploited the localized nature of degradation due to blur in image forming systems. However, due to truncated Taylor's series expansion the SRT method was found to be not suitable for restoring highly defocused images in which the support domain of the PSF is quite large. In this chapter, the multi-interval RT (MRT) approach to solve the problem of large support domain is investigated. In the MRT method, the local support domain is divided into several sub-regions and then expressed in terms of truncated Taylor's series polynomial. The focused image estimate is then recovered iteratively. In such an approach the advantages of localization and parallelizability of RT are retained, however, at increased computational cost.

The rest of the chapter is organized as follows. In section 3.1, the background and motivation for MRT approach are presented by analyzing the shortcomings of the SRT method. In section 3.2 and 3.3, detailed derivation of MRT method in 1D and 2D are presented. Section 3.4, discusses some implementation details and presents the

computational iterative algorithm for shift-variant image restoration. Section 3.5 discusses the properties and experimental results of the algorithm. In Section 3.6, a fast deblurring MRT technique is presented.

3.1. Background

In the RT based methods, due to the truncated Taylor's series expansions blurred images are expressed in terms of derivatives of focused image. Therefore, an explicitly local approach to shift-variant image deblurring is possible with RT. Taylor's series expansion central to RT based methods, however, is inaccurate for defocus with large support domains. Such inaccuracies can be reduced by including additional higher order terms in the Taylor's series polynomial of the SRT method. This approach will increase the size of the kernel matrix at each pixel, keeping all the advantages of the RT based technique. However, in practice, this is approach is not feasible as numerical differentiation of data for higher order derivatives are often corrupted by noise.

An alternative approach is to divide the local support domain into multiple smaller intervals as illustrated in Fig. 3.1. In this Figure, the largest support domain is taken to be a region of size 21×21 pixels. It is divided into 9 sub-regions of 7×7 pixels. Each sub-region is denoted by l_i, with l_0 containing current pixel at (x, y). The sub-regions also contain reference points (x_i, y_i) as shown in the figure. Following the same principle as in the SRT method, the focused image in each sub-region is modeled by a local truncated Taylor's series about its reference point. Then, a system of equations is derived for each sub-region. At each pixel, the MRT system then becomes a sum of matrix-vector products; each product term coming from a sub-region. MRT approach for 1D will be a projection of the 2D diagram shown below.

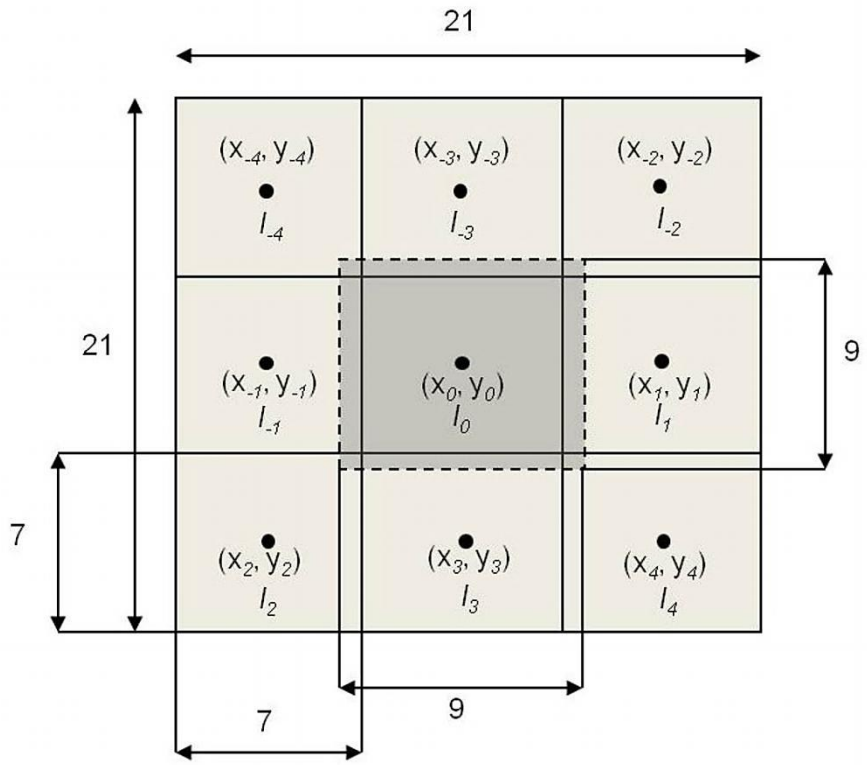


Figure 3.1: Illustrates the multi-interval RT approach for a PSF of 21×21 pixels support domain. The sub-regions are 7×7 pixels in size. A 9×9 highlighted area at the center shows the extent of the support domain of SRT method for which accurate restoration is possible.

3.2. The Multi-interval RT technique in 1D

The blur integral in 1D with blurred image g, focused image f and the shift variant PSF k is given by,

$$g(s) = \int_a^b k(s,t)\ f(t)dt \tag{3.1}$$

Following the Multi-interval idea, let the local region $a \leq t \leq b$ be split into $(2l\ +\ 1)$ non-overlapping parts. Now define:

$$k_i(s,t) = \begin{cases} k(s,t), & \text{if } a_i \le s,t \le b_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

And

$$f_i(t) = \begin{cases} f(t), & \text{if } a_i \le t \le b_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Now, Eq. (3.1) is rewritten as,

$$g(s) = \sum_{i=-l}^{l} \int_{a_i}^{b_i} k_i(s,t)\ f_i(t)dt \tag{3.4}$$

$$\Rightarrow g(s) = \sum_{i=-l}^{l} g_i(s). \tag{3.5}$$

Eq. (3.5) is the sum of *partial blur integrals* and $g_i(s)$ is the partial blur contribution from i^{th} sub-region to pixel at s. A reference point s_i is chosen such that it is within the i^{th} sub-region (usually at the center) for each sub-region. The partial blur integrals in the above equation are transformed by applying RT. Substitute $t = s_i - u$ with s_i being the reference point for that interval. After the transformation $g_i(s)$ becomes:

$$g_i(s) = \int_{s_i-b_i}^{s_i-a_i} k_i(s, s_i-u) f_i(s_i-u)\ du \tag{3.6}$$

The function $f_i(s_i-u)$ expanded in Taylor's series with respect to the reference point is given below.

$$f_i(s_i-u) = \sum_{n=0}^{N} a_n\, u^n\, f_i^{(n)}(s_i) \tag{3.7}$$

In the expression above $a_n = (-1)^n/n!$ and $f_i^{(n)}(s_i)$ is the n^{th} derivative of the function. Substituting for $f_i(s_i-u)$ in Eq. (3.6) from Eq. (3.7), the following expression is obtained.

$$g_i(s) = \sum_{n=0}^{N} a_n \; f_i^{(n)}(s_i) \int_{s_i - b_i}^{s_i - a_i} u^n \; k_i(s, s_i - u) du \tag{3.8}$$

Define the n^{th} partial moment of the PSF as,

$$k_{(i),n}(s) = \int_{s_i - b_i}^{s_i - a_i} u^n \; k_i(s, s_i - u) du \tag{3.9}$$

Differentiating Eq. (3.8) w.r.t s N times and truncating at order N a linear system is derived, as in SRT.

$$g_i^{(p)}(s) = \frac{d^p}{ds^p} \sum_{n=0}^{N} a_n \, f_i^{(n)}(s_i) \int_{s_i - b_i}^{s_i - a_i} u^n \; k_i(s, s_i - u) du \tag{3.10}$$

$$\begin{bmatrix} g_i{}^{(0)} \\ g_i{}^{(1)} \\ \vdots \\ g_i{}^{(N)} \end{bmatrix} = \begin{bmatrix} r_{i,00} & r_{i,01} & \cdots & \cdots \\ r_{i,10} & r_{i,11} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ \cdots & \cdots & & \cdots \end{bmatrix} \begin{bmatrix} f_i{}^{(0)} \\ f_i{}^{(1)} \\ \vdots \\ f_i{}^{(N)} \end{bmatrix} \tag{3.11}$$

The above equation is written in a matrix form as below.

$$\mathbf{g}_\mathrm{i}(s) = \mathbf{R}_\mathrm{i}(s)\mathbf{f}_\mathrm{i}(s_i) \tag{3.12}$$

Hence, the original SRT matrix relation is now written as a sum of matrix-vector products due to multi-interval formulation. The multi-interval matrix relation is given below,

$$\mathbf{g}(s) = \sum_{i=-l}^{l} \mathbf{R_i}(s)\mathbf{f_i}(s_i) \tag{3.13}$$

In Eq. (3.12), $i = 0$ corresponds to the sub-region at the center of the support domain of the PSF. The unknown focused image at the current pixel is in the vector $\mathbf{f}_0$. The $(k+1)^{th}$ estimate for $\mathbf{f}_0$ is found by the following update equation:

$$\mathbf{f}_0^{k+1}(s_0) = \mathbf{R}_0^{-1}(s)\left[\mathbf{g}(s) - \sum_{\substack{i=-l,\\ i\neq 0}}^{l} \mathbf{R_i}(s)\mathbf{f_i}^k(s_i)\right] \tag{3.14}$$

In the computer implementation, the iterative scheme was terminated after 4-5 iterations.

3.3. The Multi-interval RT technique in 2D

In the previous section, the Multi-interval RT technique was derived for one-dimensional signals. It will now be extended to 2D. The blur integral with blurred image, g focused image f and the shift-variant PSF k is given by:

$$g(x,y) = \int_a^b \int_c^d k(x,y,u,v) f(u,v) du\, dv\,. \tag{3.15}$$

Following the multi-interval the local region, $a \le u \le b$ and $c \le v \le d$, is splint into $(2l + 1)$ non-overlapping parts. An example of it for a 21×21 support is shown in Fig. 1. In the figure, the size of each sub-region is 7×7 with a total of nine sub regions, i.e., $l = 4$. Now, similar to 1D case, define:

$$k_i(x,y,u,v) = \begin{cases} k(x,y,u,v), & \text{if } a_i \le x,u \le b_i\,; c_i \le y,v \le d_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

$$f_i(u,v) = \begin{cases} f(u,v), & \text{if } a_i \le u \le b_i\,; c_i \le v \le d_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Eq. (3.15) is then expressed as the sum of *partial blur integrals* by using Eqns. (3.16) and (3.17).

$$g(x,y) = \sum_{i=-l}^{l} \int_{a_i}^{b_i} \int_{c_i}^{d_i} k_i(x,y,u,v) f_i(u,v)\, du\, dv\,. \tag{3.18}$$

$$= \sum_{i=-l}^{l} g_i(x,y) \tag{3.19}$$

In Eq. (3.19), $g_i(x,y)$ is the partial blur contribution from i^{th} sub-region to the blurred pixel at (x,y). Reference point (x_i, y_i) is chosen such that the point is within the i^{th} sub-region. Then, each partial blur integral in the above equation is transformed by changing variables, u and v to $u = x_i - \alpha$ and $v = y_i - \beta$ with (x_i, y_i) being the reference point for that interval. Then the partial blur contribution $g_i(x,y)$ is expressed by,

$$g_i(x,y) = \int_{x_i-b_i}^{x_i-a_i} \int_{y_i-d_i}^{y_i-c_i} k_i(x,y,x_i-\alpha,y_i-\beta) f_i(x_i-\alpha,y_i-\beta)\, d\beta\, d\alpha\,. \tag{3.20}$$

The function $f_i(x_i-\alpha,y_i-\beta)$ expanded in Taylor's series with respect to the reference point (x_i, y_i) is given below.

$$f_i(x_i-\alpha,y_i-\beta) = \sum_{n=0}^{N} \sum_{m=0}^{n} a_n C_n^m\, \alpha^{n-m}\, \beta^m\, f_i^{(n-m,m)}(x_i,y_i) \tag{3.21}$$

In the expression above, $a_n = (-1)^n/n!$ and $f_i^{(n-m,m)}(x_i,y_i)$ is $(n-m)^{th}$ and m^{th} partial derivative of the function w.r.t variables x and y respectively. Also C_n^m is the binomial coefficient of Taylor's series expansion. Substituting for $f_i(x_i-\alpha,y_i-\beta)$ in Eq. (3.20) from Eq. (3.21), the following expression is obtained.

$$g_i(x,y) = \sum_{n=0}^{N} \sum_{m=0}^{n} a_n C_n^m\, f_i^{(n-m,m)}(x_i,y_i) \int_{x_i-b_i}^{x_i-a_i} \int_{y_i-d_i}^{y_i-c_i} \alpha^{n-m}\, \beta^m\, k_i(x,y,x_i-\alpha,y_i-\beta)\, d\beta\, d\alpha \tag{3.22}$$

Define the $(n-m,m)^{th}$ partial moment with respect to variables α and β as follows:

$$k_{(i),n-m,m}(x,y) = \int_{x_i-b_i}^{x_i-a_i} \int_{y_i-d_i}^{y_i-c_i} \alpha^{n-m}\, \beta^m\, k_i(x,y,x_i-\alpha,y_i-\beta)\, d\beta\, d\alpha \tag{3.23}$$

Substituting Eq. (3.23) in Eq. (3.22) the partial blur is expressed in terms of the derivatives of the function f at the reference point (x_i, y_i).

$$g_i(x,y) = \sum_{n=0}^{N}\sum_{m=0}^{n} a_n C_n^m \; f_i^{(n-m,m)}(x_i,y_i)\; k_{(i),n-m,m}(x,y) \tag{3.24}$$

Derivatives of the above equation w.r.t x and y truncated at order N is expressed by:

$$g_i^{(p,q)}(x,y) = \sum_{n=0}^{N}\sum_{m=0}^{n} a_n C_n^m \left(\frac{\partial^p}{\partial x^p}\right)\left(\frac{\partial^q}{\partial y^q}\right) \; f_i^{(n-m,m)}(x_i,y_i)\; k_{(i),n-m,m}(x,y)\, T(n+p+q), \tag{3.25}$$

Where $p, q, n = 0,1,\dots,N$ and the truncation function T defined as:

$$T(\,n+p+q) = \begin{cases} 1, & \text{if } (n+p+q) \le N \\ 0, & \text{otherwise.} \end{cases} \tag{3.26}$$

By grouping the coefficients of $f_i^{(n-m,m)}(x_i,y_i)$ together, the *partial blur integral* in the i^{th} sub-region is written as a matrix-vector product:

$$\begin{bmatrix} g_i^{(0,0)} \\ g_i^{(1,0)} \\ \vdots \\ g_i^{(0,N)} \end{bmatrix} = \begin{bmatrix} r_{i,00} & r_{i,01} & \cdots & \cdots \\ r_{i,10} & r_{i,11} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ \cdots & \cdots & & \cdots \end{bmatrix} \begin{bmatrix} f_i^{(0,0)} \\ f_i^{(1,0)} \\ \vdots \\ f_i^{(0,N)} \end{bmatrix} \tag{3.27}$$

$$\text{or,} \qquad \mathbf{g}_\mathrm{i}(x,y) = \mathbf{K}_\mathrm{i}(x,y)\mathbf{f}_\mathrm{i}(x_i,y_i) \tag{3.28}$$

The SRT matrix relation is then expressed as a sum of matrix-vector products:

$$\mathbf{g}(x,y) = \sum_{i=-l}^{l} \mathbf{K}_\mathrm{i}(x,y)\mathbf{f}_\mathrm{i}(x_i,y_i) \tag{3.29}$$

Carrying out similar algebraic manipulations as in the 1D multi-interval case the following iterative update equation is derived.

$$\mathbf{f}_0^{k+1}(x,y) = \mathbf{R}_0^{-1}(x,y)\left[\mathbf{g}(x,y) - \sum_{\substack{i=-l\,,\\ i\neq 0}}^{l} \mathbf{R_i}(x,y)\mathbf{f_i}^k(x_i,y_i)\right] \tag{3.30}$$

The term $\mathbf{R}_0^{-1}(x,y)$ in the equation above is computed using SVD. Using SVD, Tikhonov regularization can be readily incorporated in to the method; see Sec 2.4.4.

3.4. Computer implementation of MRT

In the previous sections, mathematical formulations of the MRT method were presented. In the SRT method, PSF moments were computed through analytical expressions [53]. The partial moments in Eq. (3.23) have to be computed through numerical integration of the global form PSF k. A global form PSF is obtained from the PSF models through simple variable transformations.

3.4.1. Selecting the sub-regions

The formulation of the multi-interval method allows for the sub-regions to be rectangles of different sizes. The segmentation depends on the shape of the PSF. For defocus, since the PSFs are circularly symmetric sub-regions were of equal size as shown in Fig. 3.1.

3.4.2. Computing the PSF partial moments

The expression for computing the partial moments of PSF and its derivatives are given in Eq. (3.23). The kernel k is in global form. Therefore, the kernel at every location has to be computed through the following relation between the PSF models.

$$h(x, y, u, v) = k(x + u, y + v, u, v) \tag{3.31}$$

In the equation above, h corresponds to the PSF models discussed in Chapter 2; see Section 2.3.1. Note that the PSF σ changes for every position $(x - u, y - v)$. The partial moment is then computed using simple numerical integration techniques. Derivatives of the PSF are obtained by differentiating the PSFs w.r.t x and y. The moments of the derivatives of the PSF are then computed using numerical integration.

3.4.3. Image differentiation filters

Differentiation in 1D is carried out by centered differences derived from Taylor's series [61, 62]. For computing image derivatives, convolution differentiation filters provide an inexpensive way to estimate image derivatives. Many differentiation filters exist [64-66]. However, the choice of the filter depends on the data and its noise characteristics. Two filters were used in implementation of localized shift-variant restoration algorithms. The noise robust gradient operator (NRIGO) is provided by Holborodko[66]. It performs well under the presence of noise and thus limiting noise amplification during restoration. The x derivative is given by:

$$\frac{\partial}{\partial x}(\cdot) = (1/32)\begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix} \tag{3.32}$$

The y derivative is given by the transpose of the filter above. Higher order derivatives (up to order 3) are obtained by repeated differentiation.

The second filter computes the coefficients of a local bivariate cubic polynomial in a 5×5 region. The cubic polynomial is expressed below.

$$p(x,y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^3 + a_7 x^2 y + a_8 x_y^2 + a_9 y^3 \quad (3.33)$$

The coefficients themselves provide the values of the derivatives at the current pixel. They are obtained by convolving the image with the filters in Fig. 3.2.

The MRT method is summarized below.

The MRT method

1. *Initialize* $\mathbf{f}^k(x,y)$
2. *Compute the kernel matrices for PSFs at every pixel*
3. *Compute the blurred image derivaties*: $\mathbf{g}(x,y)$
4. *Update*: $\mathbf{f}_0^{k+1}(x,y) = \mathbf{R}_0^{-1}(x,y)\left[\mathbf{g}(x,y) \quad - \quad \sum_{\substack{i=-l, \\ i\neq 0}}^{l} \mathbf{R_i}(x,y)\mathbf{f_i}^k(x_i,y_i) \quad \right]$
5. *Compute*: $e = MSE\left(\mathrm{f}_0^{k+1} - \mathrm{f}_0^k\right)$

   if $e > T$

   $k \leftarrow k+1$

   Go to step 4.

   else

   Terminate.

Algorithm 3.1: The Multi-interval RT iterative algorithm to obtain focused image estimate, given a blurred image and 2D shift-variant PSFs

-1	2	7	2	-13
2	17	22	17	2
7	22	7	22	7
2	17	22	17	2
-13	2	7	2	-13

$$* \left(\frac{1}{175}\right) = A_0$$

31	-44	0	44	-31
-5	-62	0	62	5
-17	-68	0	68	17
-5	-62	0	62	5
31	-44	0	44	-31

$$* \left(\frac{1}{420}\right) = A_1$$

31	-5	-17	-5	31
-44	-62	-68	-62	-44
0	0	0	0	0
44	62	68	62	44
-31	5	17	5	-31

$$* \left(\frac{1}{420}\right) = A_2$$

2	-1	-2	-1	2
2	-1	-2	-1	2
2	-1	-2	-1	2
2	-1	-2	-1	2
2	-1	-2	-1	2

$$* \left(\frac{1}{70}\right) = A_3$$

4	2	0	-2	-4
2	1	0	-1	-2
0	0	0	0	0
-2	-1	0	1	2
-4	-2	0	2	4

$$* \left(\frac{1}{100}\right) = A_4$$

2	2	2	2	2
-1	-1	-1	-1	-1
-2	-2	-2	-2	-2
-1	-1	-1	-1	-1
2	2	2	2	2

$$* \left(\frac{1}{70}\right) = A_5$$

-1	2	0	-2	1
-1	2	0	-2	1
-1	2	0	-2	1
-1	2	0	-2	1
-1	2	0	-2	1

$$* \left(\frac{1}{60}\right) = A_6$$

-4	2	4	2	-4
-2	1	2	1	-2
0	0	0	0	0
2	-1	-2	-1	2
4	-2	-4	-2	4

$$* \left(\frac{1}{140}\right) = A_7$$

-4	-2	0	2	4
2	1	0	-1	-2
4	2	0	-2	-4
2	1	0	-1	-2
-4	-2	0	2	4

$$* \left(\frac{1}{140}\right) = A_8$$

-1	-1	-1	-1	-1
2	2	2	2	2
0	0	0	0	0
-2	-2	-2	-2	-2
-1	-1	-1	-1	-1

$$* \left(\frac{1}{60}\right) = A_9$$

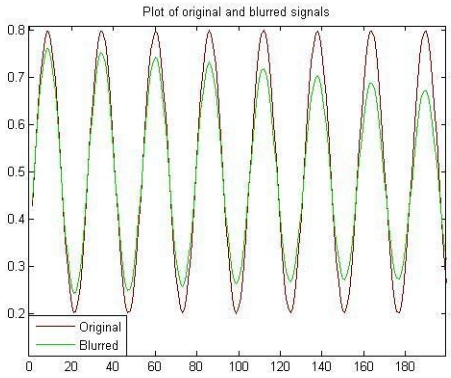
Figure 3.2: Showing different convolution filters for computing the coefficients of Eq. (1.31). Filter A_i computes the coefficient a_i. For a derivation of the filter and further discussions refer [67, 68].

3.5. Experiments and discussion

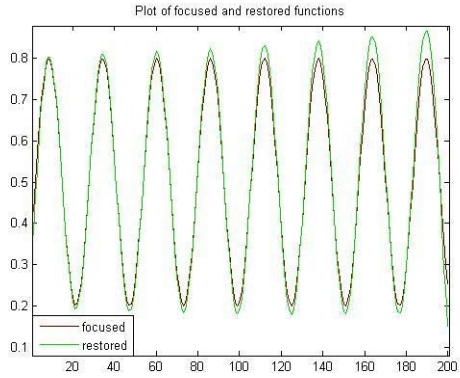
An experiment with a 1D sine function is presented first, which verifies the theory behind the MRT method. The simulation experiments demonstrate that the MRT method converges. The behavior of the MRT method is discussed with some simulation experiments. It is also compared with the SRT method in experiments.

3.5.1. Sine function example

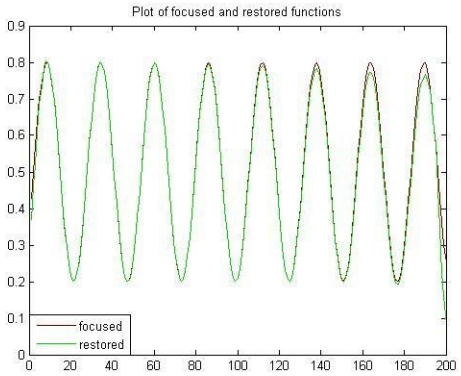
A sine wave, with a period $T = 13$, was blurred by a Gaussian PSF in 1D. The size of the blurring Gaussian kernel was 1×15. The blur parameter σ of Gaussian PSFs varied from 20% to 40% of T. In all cases, Taylor's series expansions were truncated at third order.



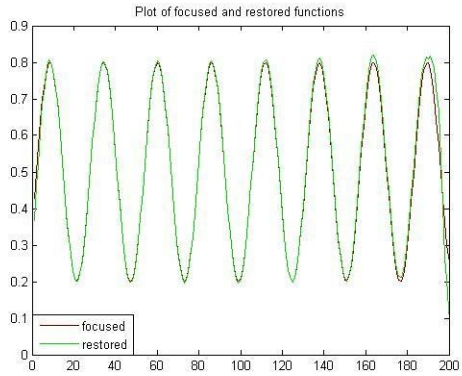
(a)



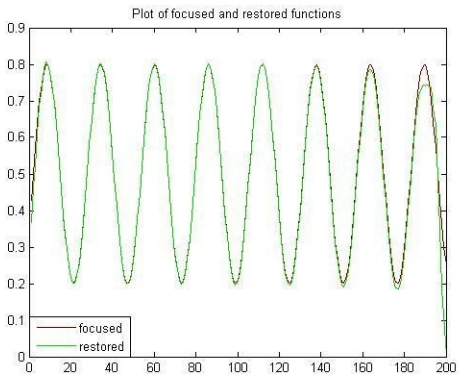
(b)



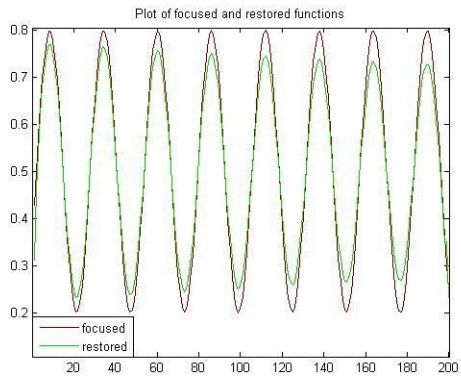
(c)



(d)



(e)



(f)

Figure 3.3: Shows the progression of solutions from MRT algorithm. (a) Sine function and its shift-variant blurred version. (b) Restoration at first iteration, RMSE=0.0229. (c) & (d) Restoration at second (RMSE=0.0110) and third iterations (RMSE=0.0147). (e) Restoration at fourth (RMSE=0.0117) and (f) SRT restoration (RMSE=0.0277)

Figure 3.3 shows clearly that the restored signal follows the original signal closely even for an analytic function and that a local truncated Taylor's series provides approximation with sufficient accuracy. The small discrepancy towards the end is attributed to errors due to boundary conditions, see Fig. 3.3.

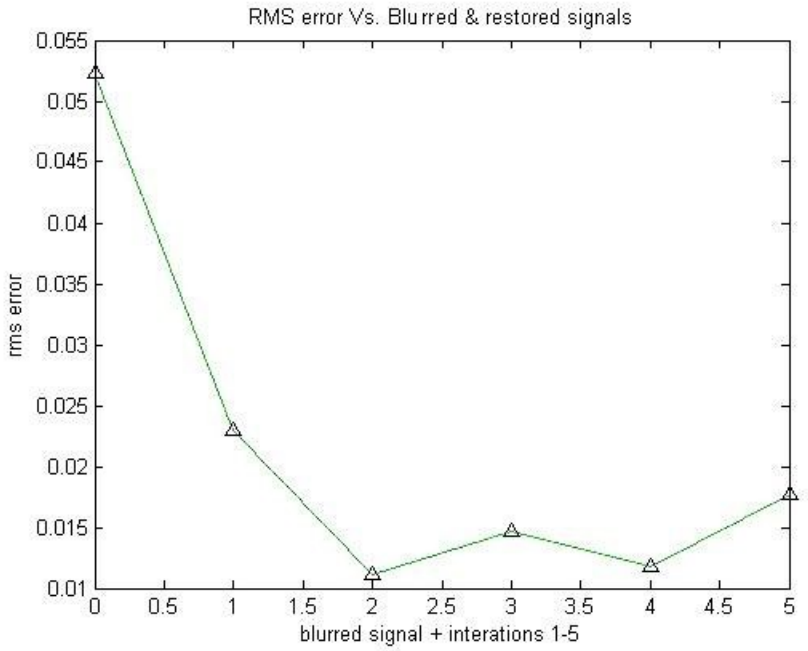


Figure 3.4: Showing RMS errors of blurred and restored signals. The 0^{th} iteration corresponds to the RMS error between blurred and original signal. The errors for iterations 1-5 show that the RMS is not strictly convex. This type of behavior was seen for iterations starting with blurred signal as the initial guess.

3.5.2. Simulation experiments

All experiments in 2D were carried out with a maximum PSF size of 21×21 pixels with each sub-region being 7×7 pixels. The MRT method was implemented in Matlab without any kind of code optimization. PSF moments were computed using simple numerical integration. In all the experiments the blur parameter was assumed to vary

linearly in image coordinates. This linear variation was used as a hash function to store and fetch PSFs based on its (x, y) location. This is an efficient way of storing shift-variant PSFs. The worst case scenario is to store the PSFs at each pixel. However, such cases do not occur often in practice. It was observed that for sufficiently smooth images regularization was not required. A regularization method such as Tikhonov regularization should be used for restoring noisy images.

The Pumpkins image in Fig. 3.5 (a) was obtained at from the URL in [46]. The image was blurred by Gaussian PSF σ increasing from 1.5 pixels at the top to 4.0 pixels at the bottom; see Fig. 3.5(b). The restored image at 4^{th} iteration with zeros as the initial solution is shown in Fig. 3.5(c). The relative MSE between the focused image and the restored solution was 0.0175. Fig. 3.5(d) shows the image in Fig. 3.5(b) restored by the SRT method. Although the overall quality of the restored image is acceptable, the image has ripple-like artifacts around step edge features in the image; see the cropped images in Fig. 3.6. This type of artifact is seen in MRT restoration as well, however, to a lesser extent. It is also noted that the relative MSE between the focused image and the restored solution was 0.05.

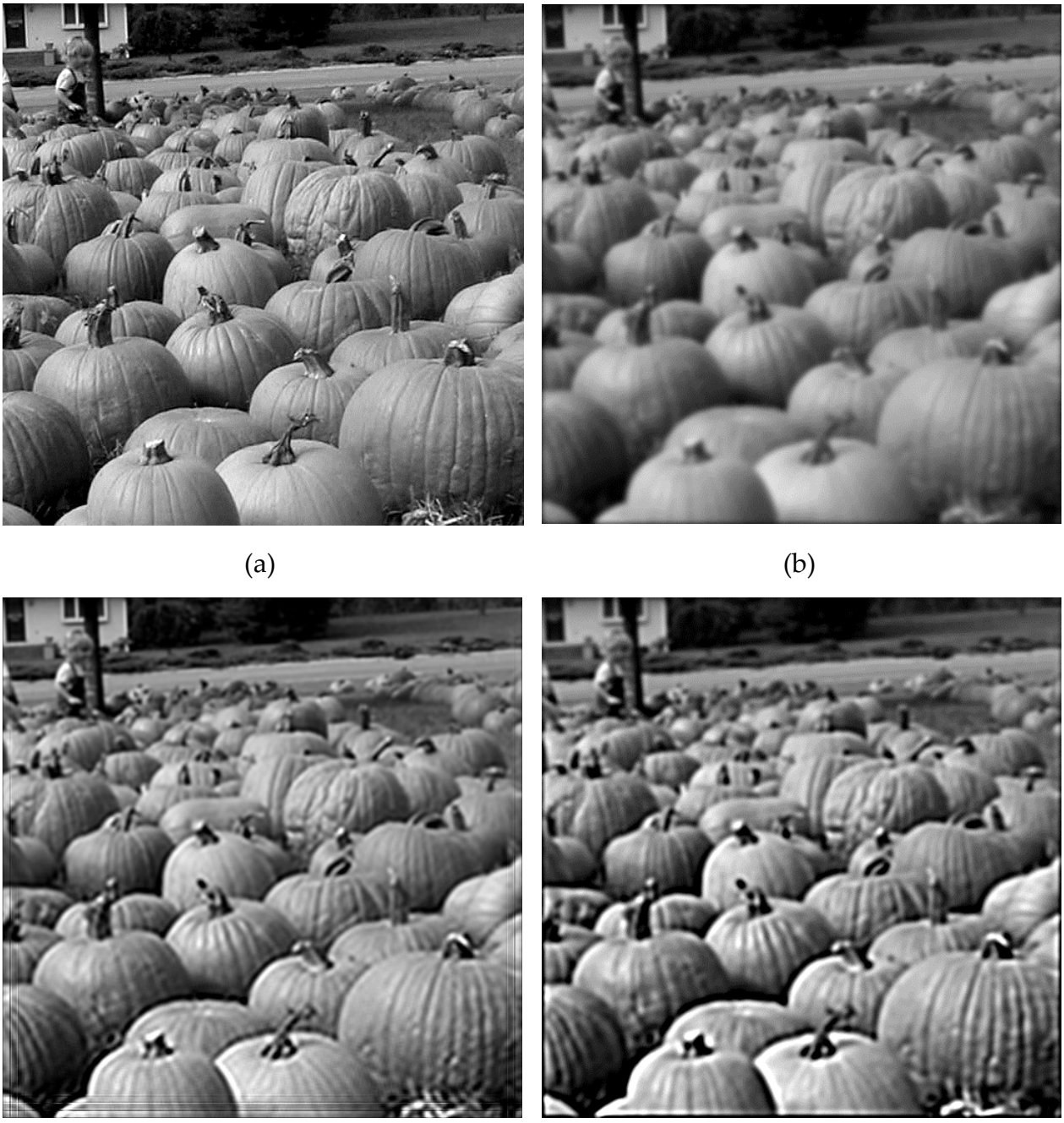
Another example is the Poster image in Fig. 3.7(a). It was acquired using Olympus C3030 digital camera by placing the poster perpendicular to the camera axis. The Poster image was blurred with Gaussian PSF σ varying from 2.0 pixels at left top corner to 4.0 pixels at right bottom corner. Fig. 3.7(b) shows the blurred image. Restored image at 5^{th} iteration with zero as the starting solution is show in Fig. 3.7(c). The relative MSE after 5 iterations was 0.0436. Fig. 3.7(d) shows the result of SRT restoration. The highlighted part in Fig. 3.8 shows that MRT restoration performs better when the underlying focused image is not smooth.

The MRT method was found to converge to a solution in about 4-5 iterations. It was also observed that the relative MSE starts growing slowly after a few iterations. The table below shows relative MSE for the two simulation cases.

Image	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Poster	0.1210	0.0515	0.0444	0.0437	0.0436
Pumpkins	0.0847	0.0233	0.0189	0.0176	0.0177

Table 3.1: Shows the relative mean squared error between the focused image and the restoration estimates at different iterations.

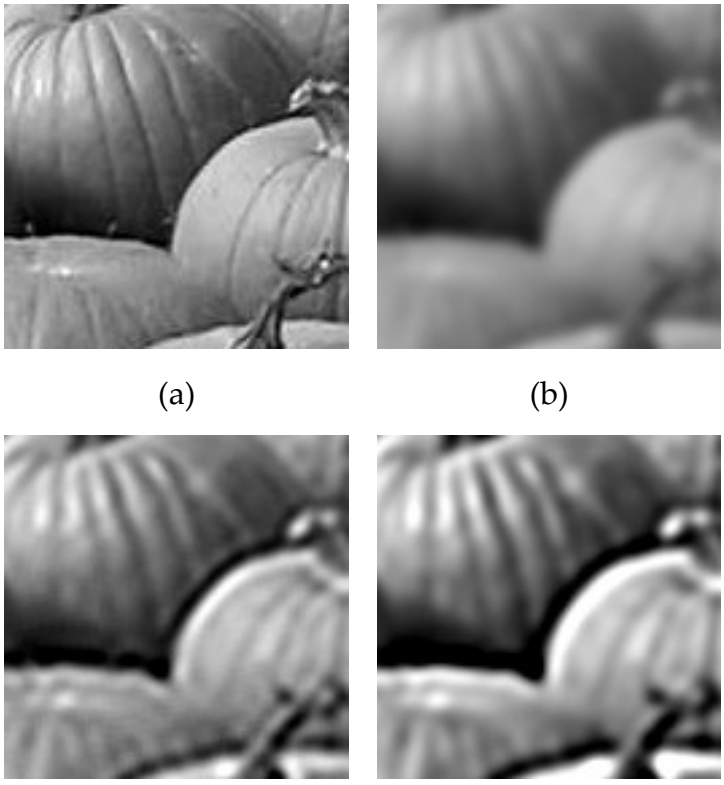
The MRT method is spent about 2 min per iteration (for 640×480 image) on an Intel core-2 PC with 2GB of RAM. The SRT method is at least twice as fast and provides the solution in one step. Some of the computational cost issues are addressed through an iterative variant of the SRT method in the next chapter.



(a) (b)

(c) (d)

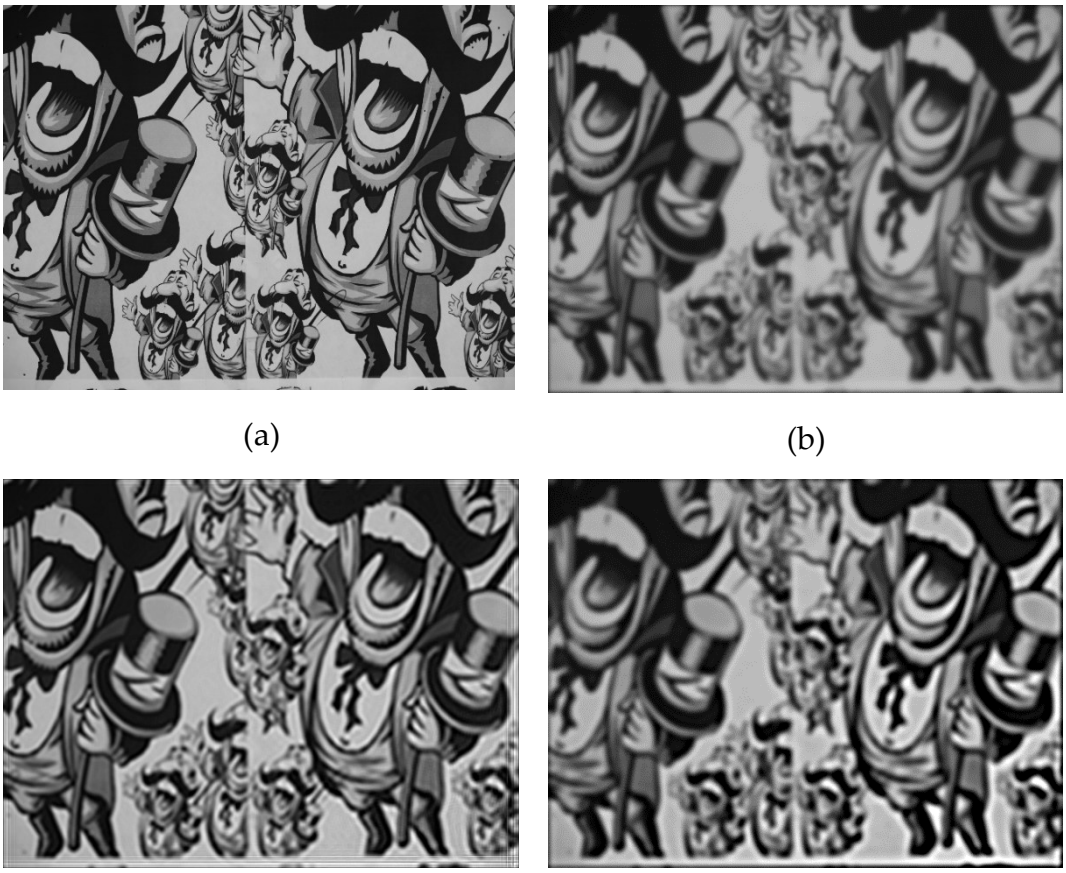
Figure 3.5: (a) Original Pumpkins image, (b) Pumpkins image blurred by shift-variant Gaussian PSF ($\sigma = 1.5 - 4$ pixels from top to bottom) and (c) Focused image estimate after 4 iterations relative MSE = 0.0175 (d) SRT restoration relative MSE = 0.05.



(a) (b)

(c) (d)

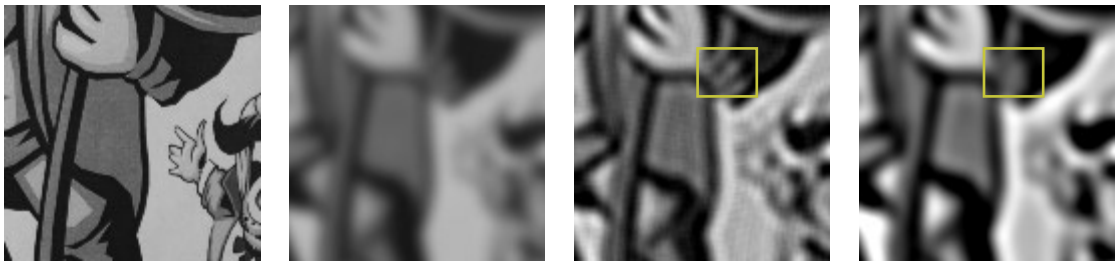
Figure 3.6: (a) Pumpkins image at original scale (cropped), (b) Blurred image, (c) MRT restoration (crop) (d) SRT Restoration (crop)



(a) (b)

(c) (d)

Figure 3.7: (a) Original Poster image, (b) Poster image blurred by shift-variant Gaussian PSF ($\sigma = 2 - 4$ pixels from top-left to bottom-right), (c) Focused image estimate after 5 iterations; relative MSE = 0.043 and (d) SRT restoration; relative MSE = 0.074



(a) (b) (c) (d)

Figure 3.8: (a) Poster image at original scale (cropped), (b) Blurred image and (c) MRT restoration (crop) (d) SRT Restoration (crop). The highlighted region shows the ineffectiveness of SRT restoration when the underlying focused image is not smooth.

3.5.3. Experiments on real images

Fig.3.9 shows results for a naturally blurred image. The Alphabet image was acquired by Olympus C3030 digital camera by placing a planar printed object inclined with respect to the camera axis. The resolution of the original Alphabet image is 640×480 pixels. For tests with Alphabet image, the PSF was assumed to be a shift-variant Gaussian function. The blur parameter σ was assumed to vary only along the horizontal direction (along x-axis or rows). It was estimated to change linearly from 4.5 pixels at the left border to 1.5 pixels near the center (column 230) and then increase back to 5.0 pixels at the right border. The results shown in Fig. 3.9(c) were obtained after three iterations, with regularization parameter value 0.1. Fig 3.9(b) shows the SRT restoration for the same parameters. It is clear that the quality of restoration is better from MRT algorithm.

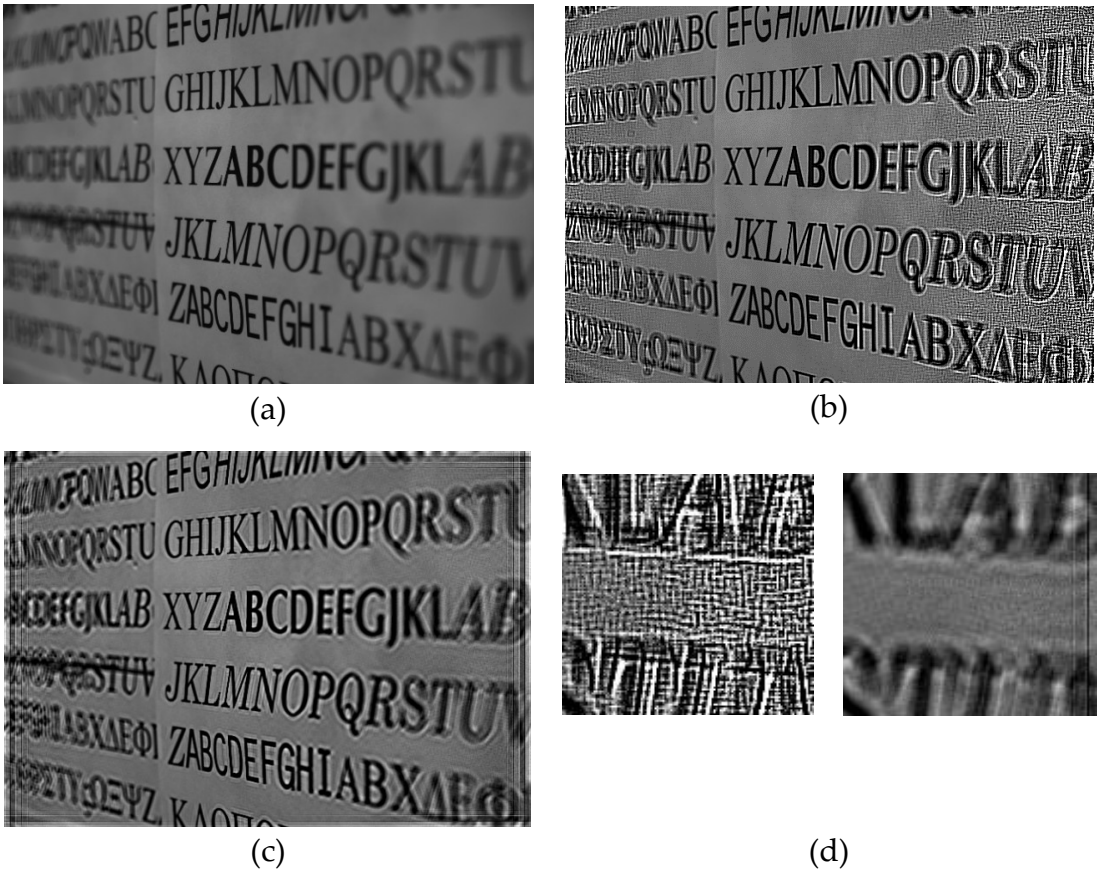


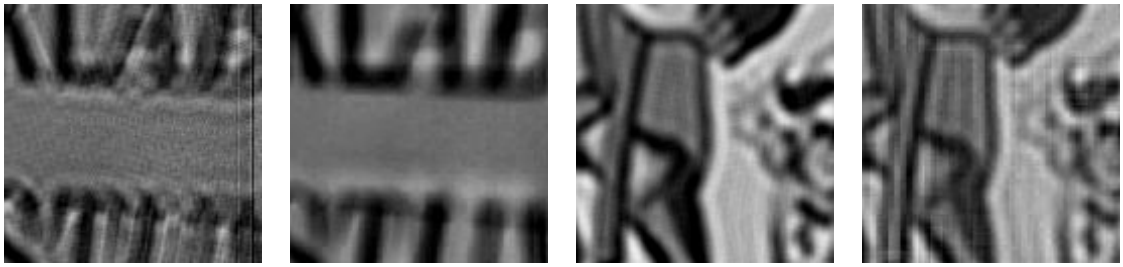
Figure 3.9: Restoration examples for real image. (a) Input blurred image, (b) SRT restoration, (c) MRT restoration and (d) Comparison of SRT and MRT restorations at 70% of original scale.

3.6. Fast deblurring by extrapolation of local solution

The solution vectors $\mathbf{f}_0(x, y)$ contain the function value f and its various partial derivatives at each pixel. This could be exploited to save computation time at the cost of some reduction in accuracy. For every $M \times M$ block centered at (x, y), the solution to all pixels can be obtained through extrapolation using Taylor's series. This way, each

$M \times M$ block requires one solution computed iteratively using the MRT technique. This fast extrapolation could result in a factor of $M \times M$ improvement in computation time.

For experiments, the blurred images were divided into 3×3 blocks, and solution was computed at the center pixel for each block. A factor of 6~7 improvement was observed in computation time. Example results are shown in comparison with the MRT method in Fig. 3.10. For the Alphabet image, a regularization parameter of 0.1 was used. The figure shows results that are visually acceptable with significant reduction in computational cost. The fast technique permits tradeoff on quality for significant improvements in computation time.



(a) (b) (c) (d)

Figure 3.10: (a) & (b) MRT and Fast extrapolation MRT restoration for Alphabet image, (c) & (d) MRT and Fast extrapolation MRT for Poster image. Relative MSE for the Poster image was 0.048

3.7. Summary

A multi-interval approach to deblur highly defocused images was presented. The MRT method iteratively recovers a solution for focused image. It was observed that the method converged after about 4-5 iterations in most cases. Through experiments, it was demonstrated that the MRT method is effective in deblurring highly defocused images that are not smooth. The MRT method also improved the accuracy of the solutions

compared with the SRT method. Using the SVD formulation, Tikhonov regularization is easily incorporated in to the method. However, the MRT method is very expensive in terms of computational cost. Most of the time is spent on the matrix-vector products, including the final product with the inverse of the central sub-kernel matrix. It is noted that alternative iterative update equations are possible with the multi-interval approach. One such example is by algebraic manipulation of Eq. (3.24) and substituting it in Eq. (3.19). This bypasses the matrix-vector formulation of the deblurring problem and hence should be more computationally efficient. A similar iterative update equation is investigated for the SRT method in the next chapter.

4. Localized and Efficient Iterative Restoration Algorithms

The MRT method for deblurring highly-defocused images was investigated in the previous chapter. It provided better restoration results than the previously discussed SRT algorithm for images that are not smooth and highly defocused. Although the MRT method was effective, it was computationally expensive compared to the SRT method. An alternative to matrix formulation for estimating the focused image is provided by iterative update equation derived from the *forward* operation equation. Since the iterative methods improve solutions gradually, they tend to provide more control over the solution.

In this chapter two localized iterative methods, based on SRT, are investigated in detail. The first method is directly derived from the *forward* RT equation and the second method simply uses the *forward* RT equation to estimate the gradient for the Landweber's method discussed in Chapter 2; see Section 2.4.2. Computational algorithms are analyzed for complexity, numerical stability and conditioning. The iterative methods also provide some new insight into the ill-conditioned nature of the deblurring problem. Deblurring noisy images is also discussed in detail. Throughout the experiments, the iterative methods are compared against the Landweber's method

and the Tikhonov regularization method (Section 2.4.4) for computation time, accuracy, robustness against noise and quality of restoration. The methods are also tested on real data.

4.1. RT iterative method derivation

The two new iterative techniques are derived from the *forward* RT equation. A detailed derivation of the SRT method is presented in Sec 2.4.5. Since the iterative techniques have similar derivation up to the forward part, the steps are skipped here to avoid redundancy. Therefore, the derivation begins from the forward RT equation.

The relation between the blurred image and the underlying focused image can be expressed through the differential equation:

$$g(x,y) = \sum_{n=0}^{N} \sum_{i=0}^{n} S_{n,i}\, f^{(n-i,i)}\ (x,y) \tag{4.1}$$

where,

$$S_{n,i} \;=\; a_n\, C_i^n \sum_{m=0}^{M} a_m \sum_{j=0}^{m} C_j^m h_{m+n-(i+j),\ i+j}^{(m-j,j)}\,. \tag{4.2}$$

In the equation above, a_n is $\frac{(-1)^n}{n!}$ and C_i^n is the binomial coefficient. Eq. (4.2) is called *forward* RT. An iterative equation for estimating the focused image is derived by algebraic manipulation of the equation above. The function value at (x,y) is isolated from the summation as,

$$g(x,y) = S_{0,0}\, f^{(0,0)}(x,y) + \sum_{n=1}^{N}\sum_{i=0}^{n} S_{n,i}\, f^{(n-i,i)}\,(x,y)\,. \tag{4.3}$$

$$S_{0,0}\, f(x,y) = g(x,y) - \sum_{n=1}^{N}\sum_{i=0}^{n} S_{n,i}\, f^{(n-i,i)}\,(x,y)\,. \tag{4.4}$$

Eq. (4.4) is used to obtain an iterative updating scheme to obtain $f_{k+1}(x,y)$ at step $(k+1)$ using the estimated solution $f_k(x,y)$ at step k as:

$$f_{k+1}(x,y) = \frac{1}{S_{0,0}}\left[\; g(x,y) \;-\; \sum_{n=1}^{N}\sum_{i=0}^{n} S_{n,i}\, f_k{}^{(n-i,i)}\,(x,y) \;\right]. \tag{4.5}$$

Eq. (4.5) recovers the focused image estimate iteratively by starting with a suitable initial guess. In the equation, the coefficient $S_{0,0}$ is usually equal to one. Therefore, with zero as the initial guess the first iteration solution is the blurred image. At every step, the estimate f is differentiated to compute the summations on the right hand side. The coefficients $S_{n,i}$ consist of the moments of the PSF. For symmetrical PSFs such as Cylindrical and Gaussian, the odd moments vanish. Hence the number of terms in the summation on the right hand side is reduced. Later, analytical expressions for these moments will be presented.

4.2. Computational algorithm and implementation

The step by step procedure for the RT iterative (RTI) technique is given by Algorithm 4.1. The method above lacks a step-size parameter. It is sometimes useful to have a step size parameter that controls the rate of convergence of an iterative technique. Therefore, as an alternative to the above equation, the Landweber's method [33] is implemented using RT to have more control. The blurring operation required in

the Landweber's method is efficiently implemented using forward RT, i.e. Eq. (4.1). The step by step procedure for implementing the Landweber's method using RT (RTLW) is presented in Algorithm 4.2.

The RT Iterative Technique

1. $Initialize\ f_{(0)}(x,y)\ and\ compute\ the\ first\ N\ moments\ of\ PSF$
2. $Compute\ partial\ derivatives\ of\ f_{(k)}(x,y)$
3. $Update: f_{(k+1)}(x,y) = \frac{1}{S_{0,0}}\Big[\ g(x,y)\ -\ \sum_{n=1}^{N}\sum_{i=0}^{n} S_{n,i}\, f_{(k)}{}^{(n-i,i)}\,(x,y)\ \Big]$
4. $Compute: e = MSE\big(f_{(k+1)}(x,y) - f_{(k)}(x,y)\big)$

if $e > T$

$k \leftarrow k+1$

$Go\ to\ step\ 2.$

else

$Terminate.$

Algorithm 4.1: RTI Algorithm for estimating focused image for a given blurred image and 2D shift-variant PSFs

The RT based Landweber Iteration

1. $Initialize\ f_{(0)}(x,y)\ and\ compute\ the\ first\ N\ moments\ of\ PSF$
2. $Compute\ partial\ derivatives\ of\ f_{(k)}(x,y)$
3. $Evaluate: \hat{g}_{(k+1)}(x,y) = \sum_{n=0}^{N}\sum_{i=0}^{n} S_{n,i}\, f_{(k)}{}^{(n-i,\ i)}\,(x,y)$
4. $Update: f_{(k+1)}(x,y) = f_{(k)}(x,y) + \alpha_k\big[g(x,y) - \hat{g}_{(k+1)}(x,y)\big]$
5. $Compute: e = MSE\big(f_{(k+1)}(x,y) - f_{(k)}(x,y)\big)$

if $e > T$

$k \leftarrow k+1$

Compute $\alpha_{(k)}$

$Go\ to\ step\ 2.$

else

$Terminate.$

Algorithm 4.2: RTLW Algorithm for estimating focused image for a given blurred image and 2D shift-variant PSFs

4.2.1. Computing PSF moments

The PSFs for Defocus aberration are modeled by 2D Cylindrical and 2D Gaussian functions. The Cylindrical PSFs are parameterized by blur circle radius R. In case of a Gaussian PSF model the spread parameter is σ and it is proportional to the blur circle radius. For simplicity only linear variation in the spread parameters is considered. That is, the blur circle radii of the PSFs are modeled by the following equation:

$$R(x, y) = R_0 + R_x\, x + R_y\, y \tag{4.6}$$

In the above equation R_x and R_y represent the slopes along x and y axes respectively. A similar equation could also be written for the parameter $\sigma(x, y)$. The moments of the PSFs are expressed in terms of the blur parameters of the PSFs. For Cylindrical PSF, the moment coefficients are computed using:

$$h_{m,n}^{(i,j)} =$$

$$\frac{(-1)^{i+j}}{\pi(m+n+2)}(i+j+1)!\ \ (R_x)^i\,(R_y)^j\,R^{m+n-(i+j)}\int_0^{2\pi} cos^m\theta\ sin^n\theta\ \ d\theta\,. \tag{4.7}$$

For a Gaussian PSF, the moments are computed using:

$$h_{m,n} = \left(\frac{1}{4\pi}\right)(\sigma\sqrt{2})^{m+n}\,\Gamma\left(\frac{m+1}{2}\right)\Gamma\left(\frac{n+1}{2}\right)[(1+(-1)^m)(1+(-1)^n)]. \tag{4.8}$$

The moments of derivatives of the Gaussian PSF are obtained by differentiating $\sigma(x, y)$ in the above expression.

Due to circular symmetry the odd moments of both the PSFs vanish. By limiting the order of Taylor's series expansion to two, Eq. (4.1) can be expanded and written explicitly as,

$$g(x,y) = f(x,y) + h_{2,0}^{(1,0)} f^{(1,0)}(x,y) + h_{0,2}^{(0,1)} f^{(0,1)}(x,y) + (1/2)\, h_{2,0}^{(0,0)} f^{(2,0)}(x,y) + (1/2)\, h_{0,2}^{(0,0)} f^{(0,2)}(x,y). \tag{4.9}$$

Similarly, for a third order Taylor's series expansion the following expression is obtained.

$$\begin{aligned} g(x,y) = {} & f(x,y) + h_{2,0}^{(1,0)} f^{(1,0)}(x,y) + h_{0,2}^{(0,1)} f^{(0,1)}(x,y) \\ & + (1/2)\, h_{2,0}^{(0,0)} f^{(2,0)}(x,y) + (1/2)\, h_{0,2}^{(0,0)} f^{(0,2)}(x,y) \\ & + (1/6)\, h_{4,0}^{(1,0)} f^{(3,0)}(x,y) + (1/2)\, h_{2,2}^{(0,1)} f^{(2,1)}(x,y) \\ & + (1/2)\, h_{2,2}^{(1,0)} f^{(1,2)}(x,y) + (1/6)\, h_{0,4}^{(0,1)} f^{(0,3)}(x,y)\,. \end{aligned} \tag{4.10}$$

Eqns. (4.9) and (4.10) are used in step 3 of the iterative algorithms. Note that $S_{0,0} = 1$ in Eq. (4.9). This is due to the requirement of PSF to conserve energy. For cases when the change in PSF is abrupt or if the PSFs are not smooth, the Taylor's series expansion above may not be suitable. Then, the moments have to be estimated through numerical integration. Note that moments are required to be computed only once in the preprocessing step of the iterative algorithms.

4.3. Analysis

A discussion of the different features of RT based iterative techniques is presented here. The algorithm is analyzed in terms of its computational complexity. Some comments are provided on its parallelizability and stability.

4.3.1. Computational complexity

For the purposes of this discussion an image of size $p \times q$ is considered. The total number of pixels in the image is $N = pq$. The iterative computations in both RTI and RTLW algorithms can be divided into two main steps.

The first step is the estimation of derivatives of the current solution. Differentiating an image is a convolution operation; see Section 3.4.3. Therefore it is possible to use FFT to estimate the derivatives in $O(NlogN)$ operations or lesser. If the size of the differentiation filter is less than $logN$ then the spatial domain implementation is possible with $O(kN),\ k < log\ N$.

The second step is to update the focused image estimate. Since RT expresses blurred image as a weighted sum of focused image and its derivatives, there are M operations per pixel or, $O(MN)$ operations for the entire image. Note that $M \ll N$ and $M < logN$. For Defocus blur many of the weights vanish making $M = 4$, see Eq. (4.9), which is quite less than $logN$ for most images.

Therefore, the number of operations per iteration is $O(NlogN + MN + C) \approx O(N\ logN)$.

4.3.2. Parallelizability

Due to localization of the blur integral equation, the RT method makes parallelizability of computation explicit. Convolution operation of differentiating the focused image estimate is easily parallelized if implemented in the spatial domain. The

second step involves finding a sum of products at every pixel, which also is straightforward for a parallel implementation.

4.3.3. Numerical stability and conditioning

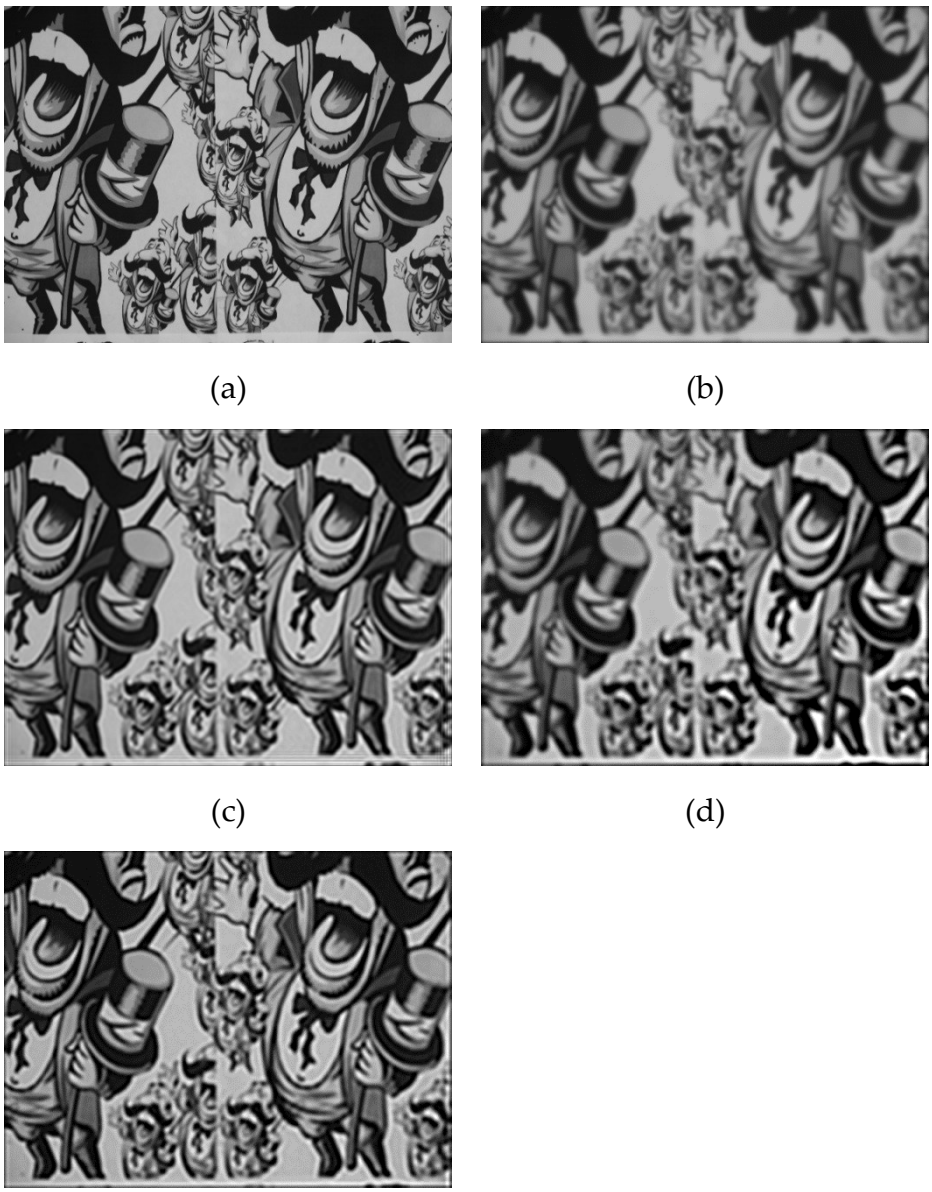
It is well known that ill-posed inverse problems are solved using some form of regularization. By choosing smooth differentiation filters and truncated Taylor's series, RT based methods implicitly perform regularization during restoration.

An interesting observation can be made from the RT iteration update equation. It is that, in updating the focused image estimates the derivatives (which amplify noise) are multiplied by the PSF moments; see Eqns. (4.9) and (4.10). The PSF moments are proportional to a power of the blur circle parameter. In fact, roughly the n^{th} moment is proportional to the n^{th} power of the blur circle parameter. Therefore with increasing blur, the moments grow rapidly and in turn multiply the derivatives. This provides a localized perspective to the conditioning of the inverse problem. That is, in regions of the image affected by large blur the inversion is more sensitive to noise amplification. Therefore it is imperative that the noise in the derivative estimates be suppressed more in regions where there is large blur. In extreme cases, it may be necessary to truncate the derivative estimates to keep the errors in check.

4.4. RT iterative, MRT and SRT methods

In this section, a brief comparison of RT iterative methods is made with the MRT and the SRT methods. The Poster image, shown in Fig. 4.1(a), was blurred with Gaussian PSF σ varying from 2.0 pixels at left top corner to 4.0 pixels at right bottom

corner. The blurred and the restored images from various RT based methods are shown in Figs. 4.1(b)-(e). The cropped images of restoration in Fig. 4.2 show that the MRT method is slightly better in terms of accuracy and quality of restoration. However, the RTI technique recovered the restored images in five iterations and the total time spent was about 3.6 seconds. The MRT method on the other hand spent about 120 seconds per iteration. The comparisons only provide a broad idea of computational complexity of the methods as the time measurements were for unoptimized Matlab implementations. In general, the RT iterative method is faster as it has to perform only sum-of-products operation at every pixel instead of matrix-vector products. The same argument holds for comparing SRT and RT iterative methods. Additionally, the SRT methods are known to be not suitable for large blur and the RT iterative method may perform better compared to SRT method for highly defocused images.



(a) (b)

(c) (d)

(e)

Figure 4.1: Showing original, blurred and restored Poster image. (a) Original Poster image, (b) Blurred image, (c) MRT restoration (ReMSE = 0.043), (d) SRT restoration (ReMSE = 0.074) and (e) RT iterative restoration (ReMSE = 0.050).

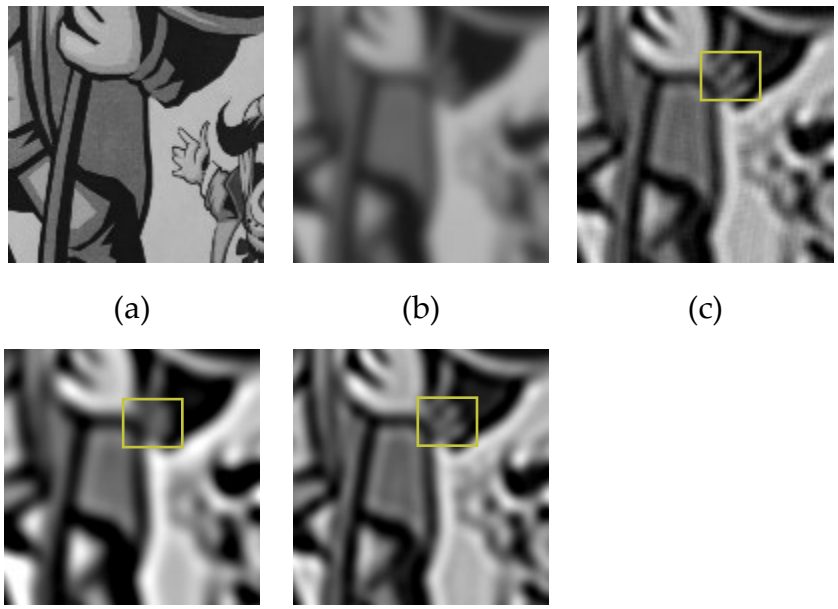


Figure 4.2: Showing cropped parts from Fig. 4.1. (c), (d) and (e) correspond to MRT restoration, SRT restoration and RT iterative restoration respectively. The highlighted part shows that MRT method is able to recover the underlying structure better than SRT and RT iterative methods.

4.5. Comparison with other deblurring methods

In this section RT iterative method and RT based Landweber's method are compared with modified Landweber's method [33] and Tikhonov regularization using SVD [63]. For comparison, the implementation of SVD available on the internet [46] was used. The table below denotes the abbreviations used in plots and the discussion below.

Name	Description
RT	RT based method (Algorithm 1)
RTLW	RT based Landweber iteration (Algorithm 2)
LW	Modified Landweber iteration
Tikh32	32x32 blocked Tikhonov regularization (SVD)
Tikh64	64x64 blocked Tikhonov regularization (SVD)

The methods are compared both quantitatively and qualitatively. For quantitative comparison relative mean squared error (ReMSE) was chosen as the metric to determine accuracy.

$$ReMSE = \frac{\sum(\hat{f} - f)^2}{\sum f^2}$$

Two kinds of quantitative comparison experiments were carried out. First set of experiments measured the computation time (in seconds) to obtain the most accurate estimate for the focused image. The second set of experiments focused on a comparison of noise sensitivity of the different deblurring algorithms. Gaussian noise was added at different SNRs, ranging from 30dB to 10dB, and the best ReMSE of the outputs of different techniques were recorded.

4.5.1. Computation time comparisons

The Sine image in Fig. 4.4 was blurred with Cylindrical PSFs with radii increasing linearly from left to right. The minimum radius was 1.5 pixels and the maximum radius was 11.0 pixels. The size of the blurred images considered varied from 128×128 to 1024×1024. The different shift-variant restoration techniques were run on a PC with Intel's Core 2 processor with 2.13 GHz clock speed and 2 GB of RAM. The algorithms were implemented in Matlab. The numbers give a general idea of the computation time requirements of different methods without any kind of code optimization. The time to obtain best estimate is tabulated below.

	Pixels	RT		RTLW	LW	Tikh32	Tikh64
Computation time (ratio with RT)	128×128	0.0364s	1	2.54	177.39	4.40	2.21
	256×256	0.1965s	1	1.61	136.00	5.40	1.66
	512×512	0.6316s	1	2.28	168.05	5.30	2.55
	1024×1024	2.4090s	1	1.91	175.58	5.49	2.22

Table 4.1. Measured time (in seconds) to generate best ReMSE estimate for blurred images of different size

The table clearly shows the computational efficiency of RT. It was observed that the RT method converges to the best solution within three-four iterations on average. The RT based Landweber's (RTLW) method behaved almost exactly as the Landweber's method, see Figure 4.3. It compares the typical behavior of RT based methods against Landweber's (LW) method. The restoration results of various techniques are shown in Fig. 4.4. Careful inspection of the solutions obtained from Tikhonov regularization show blocking artifacts. It is noted that block size has to be chosen through trial and error to obtain the best results. On the other hand, the methods proposed in this paper produce no visible blocking artifacts.

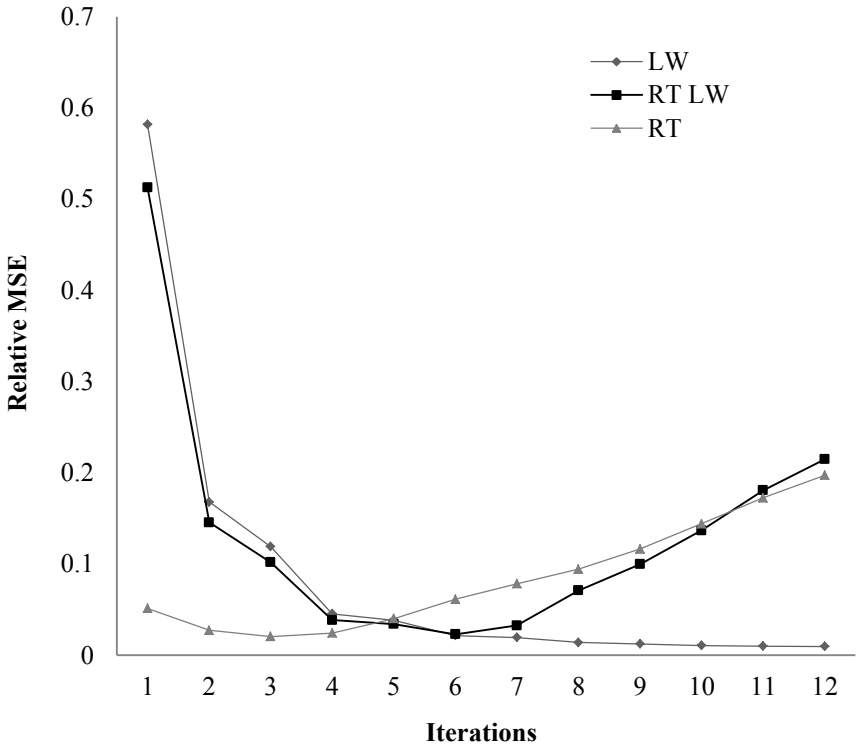


Figure 4.3: Showing a typical pattern of ReMSE error versus iterations of RT and RTLW algorithm compared against LW algorithm.

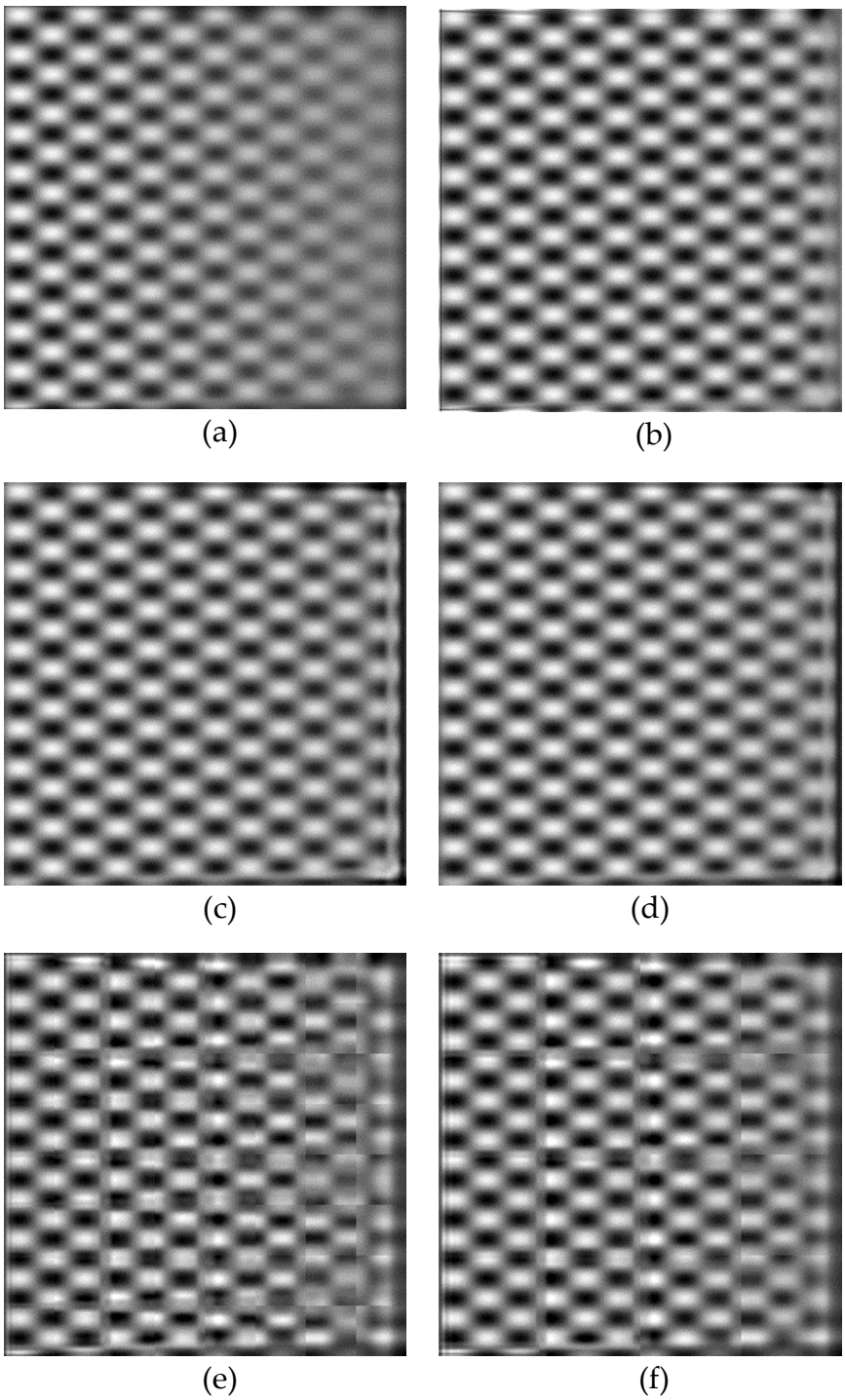


Figure 4.4: Showing restoration for Sine image, (a) Blurred image, (b) Restoration by LW method (c) RTLW method, (d) RT method, (e) Tikh32 and (f) Tikh64.

4.5.2. Restoration of noisy images

Several images were blurred with both Cylindrical and Gaussian shift-variant PSFs. The blurred images were then degraded with different amounts of noise. The amount of noise considered for the experiments were in the 0.1 to 10 percent range of signal power. This is indicated in the plots through SNR from 10dB to 30dB.

The blurred images were computed at double precision, digitized and stored, hence also adding quantization noise. The noisy blurred images were smoothed with a small Gaussian filter before feeding into the restoration algorithms. This was done to keep noise levels at a minimum in iterative techniques as they do not regularize noise amplification. The RT based methods implicitly minimize noise amplification at the time of computing the derivatives. The application of a noise suppressing derivative filter is critical to obtain satisfactory results from RT based methods in the presence of noise.

Figure 4.5 suggests that the LW method estimates the solution with highest accuracy. The RT based methods are slightly more accurate compared to Tikhonov regularization methods. The improvement in accuracy over Tikhonov regularization depended on the local smoothness of images. The RT based methods are expected to behave this way as they approximate the focused image through a truncated Taylor-series polynomial. Tikhonov regularization loses accuracy by assuming shift-invariance in small blocks. For applications where accuracy is critical the estimate obtained through RT could be used as the starting solution for the LW method. This should greatly speed up the process of obtaining a more accurate estimate to focused image. Figures 4.6 and 4.7 show the restored images obtained for a 20dB SNR and a 10dB SNR case respectively. The plots in Figure 3 indicate that the RT based methods work quite

satisfactorily up to 20dB SNR. It appears that more noise suppression in computation of the derivatives is required to obtain better results at higher levels of noise, that is, when SNR is less than 20dB. It is also observed that Tikhnov regularization performs well under large amounts of noise.

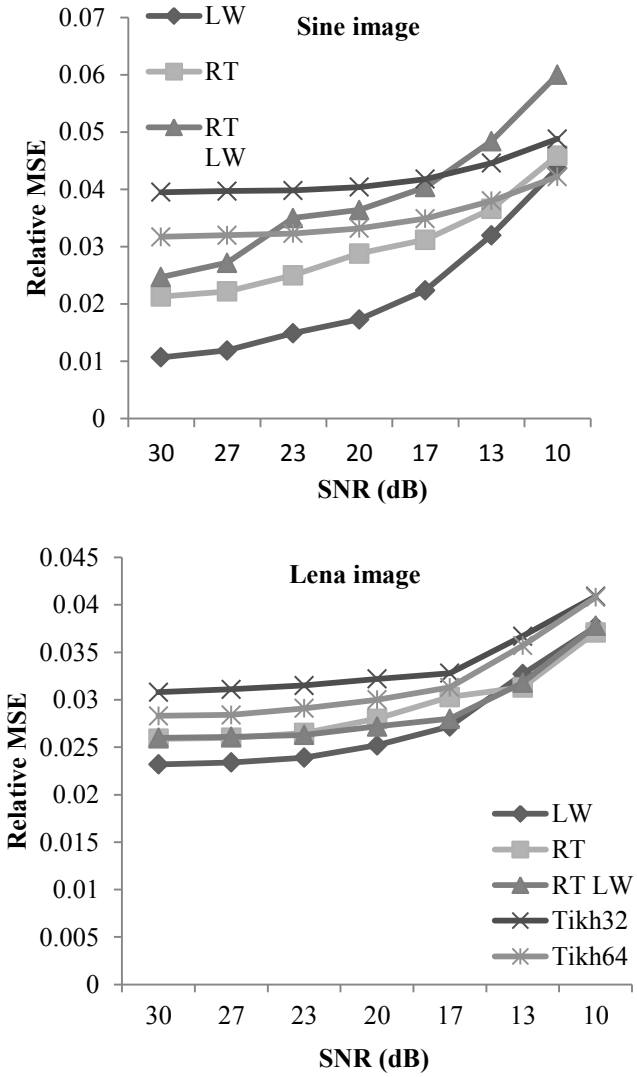


Figure 4.5: Noise sensitivity plot for Sine image (top) and Lena image (bottom). The amount of noise roughly doubles every step on the horizontal axis.

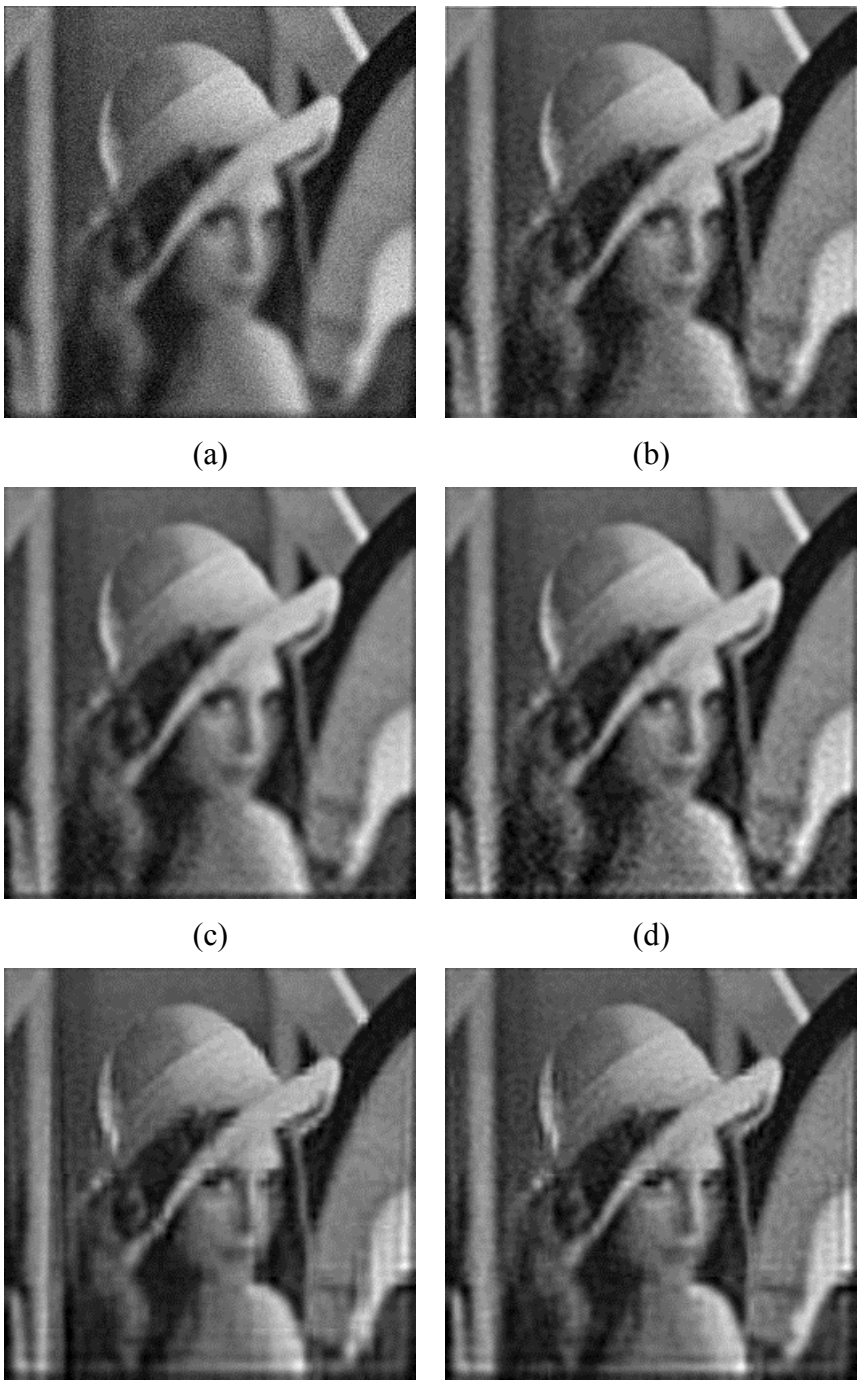


Figure 4.6: Noisy Lena image restoration. (a) Blurred Lena image (with Cylindrical SV-PSFs, radius: 1.5 pixel at the top - 4 pixels at the bottom) at 20dB SNR, Restoration by (b) LW method, (c) RTLW method, (d) RT method, (e) Tikh32 and (f) Tikh64.

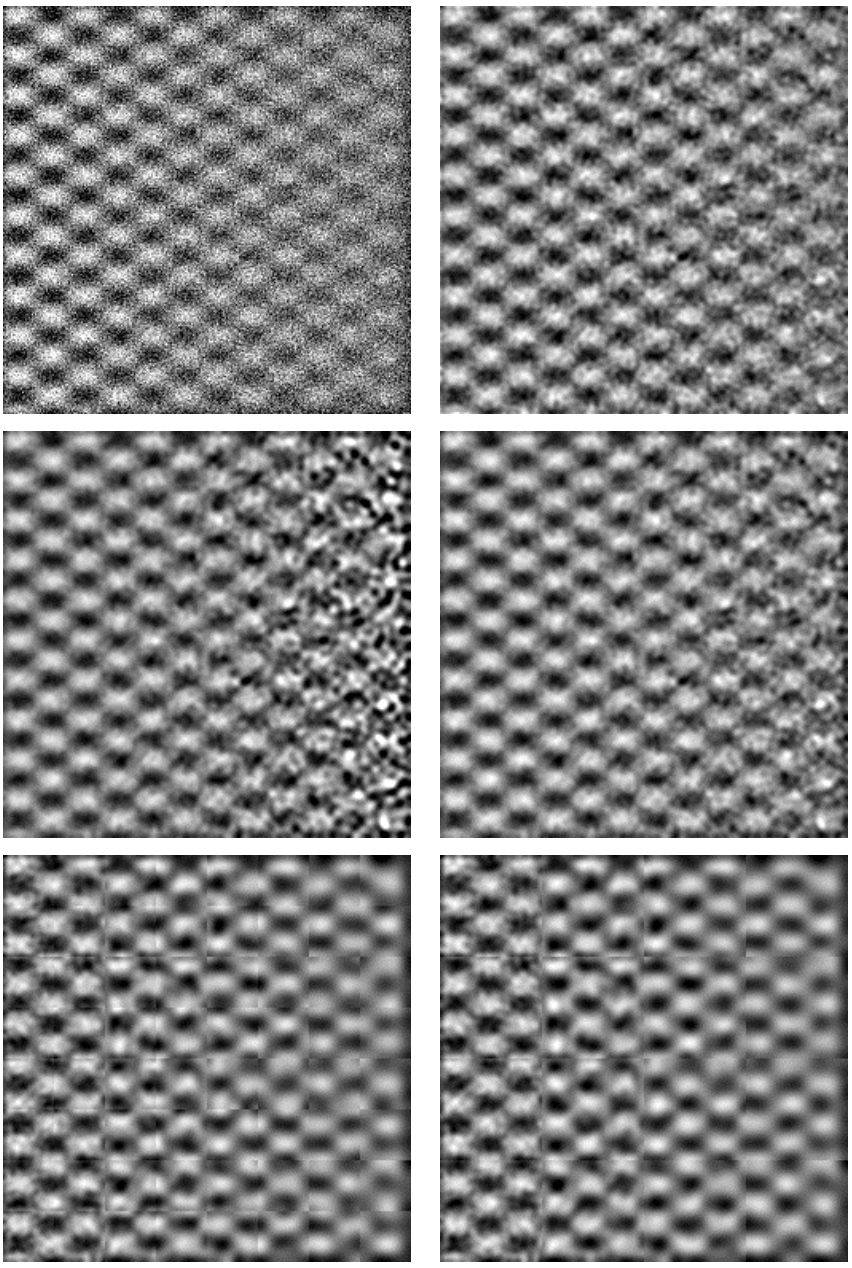


Figure 4.7: Order: Left to right and top to bottom, Blurred Sine image (with Gaussian SV-PSFs, sigma: 1.5-6 pixels) at 10dB SNR, restoration by LW method, RTLW method, RT method, Tikh32 and Tikh64.

4.5.3. Qualitative comparison

The blocking artifacts observed with sectioning methods are discussed here. The figure below highlights the region in the image that is compared at original scale. Both Pumpkin and Sine images shown are obtained from RT iterative method. To the right of the scaled image is the highlighted part (128×128 crop) at original scale. The bottom part of the figure shows 128×128 pixels taken from the restored images obtained from Tikh32 and Tikh64 methods respectively. It was noted that the sectioning methods produce blocking artifacts. In some cases, such as the Sine image the blocking artifacts were visible even without careful inspection. On the other hand, the iterative methods did not introduce blocking artifacts in any region.

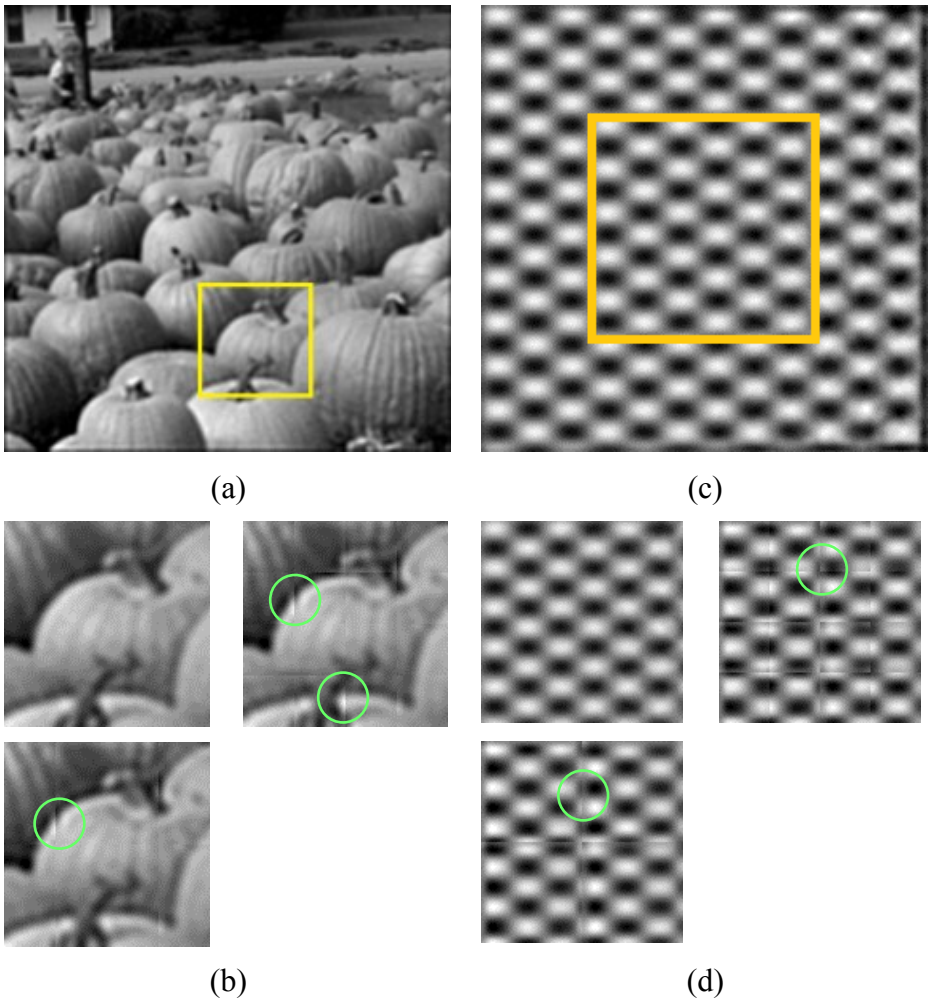


Figure 4.8: Showing blocking artifacts produced in sectioning methods for image restoration such as SVD based Tikhonov regularization. (a) Pumpkin image restored by RT iterative method and (b) qualitative comparison of RT and Tikhonov methods. (c) Sine image restored by RT iterative method and (d) qualitative comparison of RT and Tikhonov methods.

4.5.4. Experiment on real data

The details of experiments on real data are presented here. Images of different planar objects were taken such that they were inclined at an angle to the optical axis. The RT based shape recovery algorithm [10] was used to acquire the information about the shift-variant PSFs. Another shape recovery technique based on Depth from Defocus

can also be used to estimate the shift-variant PSFs [11]. The RT based technique suits our algorithm as it estimates the blur parameter (e.g., radius) by fitting a plane. The blur parameter at a position (x, y) can be easily obtained by evaluating the equation of the plane.

For the experiments images were acquired through a computer controlled Olympus C3030 digital camera. The objects were placed between 50 cm to 100 cm from the camera. The estimates generated through RT based shape recovery technique were used to compute the moments of shift-variant PSFs and restore the blurred images. Figure 4.9 shows an example of shift-variant image restoration using a Cylindrical PSF model for a real image. The blurred input image is shown in Figure 4.9(a). Figures 4.9(b) and 4.9(c) show the result of RTI algorithm and the actual focused image. The outputs of different restoration algorithms taken from the highlighted region in respective images are shown in Figure 4.9(d) at original scale. It is evident from the figure that the output of the RT based technique closely matches the focused image. Tikhonov regularization produces some artifacts. RT based techniques produced the restored result fastest followed by Tikhonov regularization and LW method. This example demonstrates that the RT based techniques provide good quality restoration results at minimum computational cost.

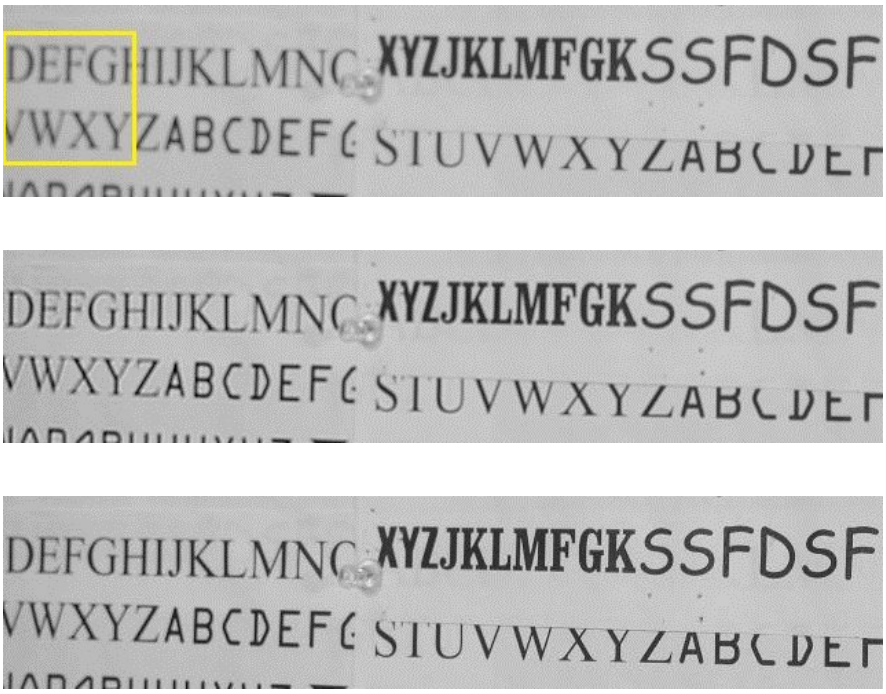


Figure 4.9: An example of RT iterative implementation for real data, (a) Blurred image (estimated radius 2.0075 at the left end, 0.559 at the right end), (b) Restored by RT iteration technique; ReMSE = 0.0101 with (c), and (c) Actual focused image (captured at a different camera setting)

Figure 4.10: Restoration samples from different techniques (at original scale). Order: left – right and top – bottom. Blurred image, focused image, restoration by LW method, RTLW method, RT method and Tikh32.

4.6. Conclusions

Two new RT based iterative techniques were presented for shift-variant image restoration. It was demonstrated that the techniques have significant computational advantage over other iterative methods. The techniques perform well for smooth images. The accuracy of solution is sufficient but not as much as the Landweber's method. The algorithms perform implicit regularization and provide robustness against low to medium levels of noise. A new insight into the ill-conditioned nature of the shift-variant image restoration problem was also discussed. Based on this, it may be possible to perform localized regularization in the spatial domain at regions blurred by large amounts of defocus. The current investigation indicates interesting possibilities in this direction. Another extension of this work is possible with restoration of shift-variant motion blur and other shift-variant aberrations.

5. Generalized Convolution Theorem: Proof and Verification

Many physical processes, including image formation, are modeled by the Fredholm Integral Equation of the First Kind [69, 70]. Numerically evaluating this integral through quadrature is generally computationally expensive. The order of computations is $O(N^2)$ when the kernels are not localized. In the previous chapters, the computational complexity was brought down due to local support domain of kernel $h(s,t)$. In this chapter, a new theorem called the Generalized Convolution Theorem (GCT) is investigated. GCT was first proposed and proved by Dr. M. Subbarao (see US patent application filed in August 2011 by M. Subbarao, S. Sastry, and S. Dutta). GCT is useful in transforming the Fredholm Integral Equation of the First Kind to a Convolution integral. In other words, GCT transforms some linear shift-variant (LSV) system to a linear shift-invariant (LSI) system. The purpose of such transformation is to make use of the Fast Fourier Transform (FFT) to reduce the computational complexity of a computer implementation of LSV systems and their inverses. The transformation is possible when the kernel $h(s,t)$ is in a suitable form. GCT provides a general framework to describe the form of the kernels under which it is possible to transform an LSV system to an LSI system.

This and the next chapter explore the possibility of reducing the computational complexity of Eq. (5.1) even when the kernels do not have a local support domain. The rest of the chapter is organized as follows. A brief background on numerical computations of an LSV system is presented. The GCT is then stated and proved in Section. 5.2. Computational algorithms are developed to apply GCT to LSV systems in Section 5.3. In section 5.4, some numerical examples are presented that verify the GCT.

5.1. Background

A linear shift-variant system is modeled by the following equation.

$$g(s) = \int_a^b h(s,t)f(t)dt \quad \text{for} \quad a \leq s,t \leq b \tag{5.1}$$

For kernels that do not have limited support domain, the systems of the form in Eq. (5.1) are computed using a matrix – vector multiplication, as shown below.

$$\mathbf{g} = \mathbf{H}\,\mathbf{f} \tag{5.2}$$

The above linear system is obtained by discretizing Eq. (5.1). Its computational complexity is $O(N^2)$. The element on the i^{th} row and j^{th} column of matrix $\mathbf{H}$ are denoted by $h_{i,j}$. Now, let s and t in Eq. (5.1) vary as,

$$s_i, t_i = a + \frac{(i-1)}{(N-1)}\,(b-a)\,.$$

Then the following relation is obtained through two point quadrature of Eq. (5.1).

$$h_{i,j} = h(s_i, t_j)\frac{(b-a)}{(N-1)} = h(s_i, t_j)\,\Delta t\,. \tag{5.3}$$

Now, Eq. (5.2) is an approximation to the continuous integral in Eq. (5.1)

5.2. GCT: Statement and proof

Statement: Let

$$g(s) = \int_a^b h(s,t)f(t)dt \text{ for } a \leq s,t \leq b\,. \tag{5.4}$$

Also, let there exist one-to-one and onto functions f_1, f_2 defined on [a, b] with continuous first derivatives such that their inverses f_3, f_4 are also one-to-one and onto, and have continuous first derivatives. Let $c = f_2(a)$, $d = f_2(b)$ then,

$$g\big(f_3(u)\big) = \int_c^d h_1(u-v)f(f_4(v))dv, \tag{5.5}$$

if and only if $h_1(u-v) = h(f_3(u), f_4(v))$ for all $u \in [f_1(a), f_1(b)]$ and $v \in [c,d]$.

Proof: The theorem states that the following condition must be satisfied to transform Eq. (5.1)

$$h_1(u-v) = h(f_3(u), f_4(v)) \tag{5.6}$$

Let:

$$g_1(u) = g(f_3(u)) = g(s) \tag{5.7}$$

$$f_5(v) = f(f_4(v)) = f(t) \tag{5.8}$$

$$f_6(v) = f_5(v)\frac{df_4(v)}{dv} = f_5(v)f_4{}'(v) \tag{5.9}$$

Note that

$$dt = \frac{df_4(v)}{dv}dv \equiv f_4{}'(v)\,dv \tag{5.10}$$

$$g(s) = g_1\big(f_1(s)\big) \tag{5.11}$$

$$f(t) = f_5\big(f_2(t)\big) \tag{5.12}$$

Now consider Eq. (5.4), from Eqns. (5.7), (5.8), (5.9) and (5.10) it is rewritten as,

$$\Rightarrow g_1(u) = g\big(f_3(u)\big) = \int_{c=f_2(a)}^{d=f_2(b)} h\big(f_3(u), f_4(v)\big)\; f(f_4(v))\; f_4{}'(v)\,dv$$

$$= \int_c^d h\big(f_3(u), f_4(v)\big)\, f_5(v)\, f_4'(v)\, dv$$

$$\Rightarrow \int_c^d h\big(f_3(u), f_4(v)\big)\, f_6(v)\, dv = \int_c^d h_1(u-v)\, f_6(v)\, dv$$

Since this is true for all f, Eq. (5.6) is satisfied if the above equation is satisfied. By changing variables $s = f_3(u)$ and $t = f_4(v)$ the Eq. (5.4) becomes:

$$g\big(f_3(u)\big) = \int_c^d h\big(f_3(u), f_4(v)\big)\ f\big(f_4(v)\big)\, f_4'(v)\, dv\ . \tag{5.13}$$

Making use of the condition in Eq. (5.6) and by substituting Eqns. (5.7), (5.8) and (5.9) in appropriate places in Eq. (5.13) the following is obtained.

$$g_1(u) = \int_c^d h\big(f_3(u), f_4(v)\big)\ f_6(v)\, dv = \int_c^d h_1(u-v)\, f_6(v)\, dv \tag{5.14}$$

Hence the Generalized Convolution Theorem is proved.

5.3. Computational algorithm

In this section the computational algorithms to apply GCT to a system of the form in Eq. (5.1) are presented. The further discussions will refer to Eq. (5.2) as the LSV system unless otherwise stated. Evaluating an LSV system by computing the matrix – vector product is an $O(N^2)$ operation. This is generally referred to as the *forward* operation. Finding the unknown **f** with a given **H** and **g** is known as the *inverse* operation. GCT is useful in computing both *forward* and *inverse* operations. Evaluating $h_1(u-v)$ is a necessary step in both *forward* and *inverse* operations, using GCT. The following algorithm provides one of the ways $h_1(u-v)$ could be computed.

Computing the transformed domain kernel h_1

1. Let $u = c; \Rightarrow s = a$
2. Compute $m_0 = min(u - v) \forall\, u, v \in [c\,,\ d]$; $m_0 = c - d$, if $c < d$.
3. Compute k^{th} sample $h_1(p_k)$, $p_k = u - v = m_0 + (k - 1) * T_0$, where $k = 1, \ldots, 2N$, $T_0 = \Delta v = \frac{(d-c)}{(N-1)}$
4. Compute $t = f_4(u - p)$; $h_1(p) = h(s,t)\frac{\Delta v}{\Delta t}$.

Algorithm 5.1: Algorithm to compute the transformed domain kernel $h_1$from the shift-variant kernel $h(s,t)$

For computing the *forward* operation, it is necessary to evaluate $f_6(v)$ from $f(t)$. The kernel $h_1(u - v)$ is computed from Algorithm 5.1. Note that $h_1(u - v)$ has $2N - 1$ samples. The integral in Eq. (5.14) is linear convolution and appropriate zero padding must be used for computing the integral using FFT. The computational steps necessary to apply GCT for *forward* operation is as follows:

Forward operation using GCT

1. Interpolate $f_6(v)$ from $f(t)$ using Eqns. (5.8) and (5.9)
2. Compute $h_1(u - v)$ using Eq. (5.6); See Algorithm 5.1
3. Compute $G_1(w) = F_6(w).H_1(w)$
4. Using inverse FFT, compute $g_1(u) \overset{DFT}{\Longleftrightarrow} G_1(w)$
5. Transform $g_1(u)$ back to the original domain using Eqns. (5.7) and (5.11)

Algorithm 5.2: Algorithm to compute the blurred image in the original domain, given a focused image f and the shift-variant kernel $h(s,t)$ through GCT.

For computing the *inverse* operation through GCT the observed signal g is first transformed to the new domain using Eq. (5.7) through some interpolation method. The kernel in the new domain is computed using Algorithm 5.1. Then an estimate for the unknown in the new domain $\hat{f}_6$ is obtained through a Deconvolution algorithm. Finally, the estimate is brought to the new domain through Eqns. (5.9) and (5.8) with the help of interpolation. The computational algorithm for finding the unknown f is as follows.

Inverse operation using GCT

1. Transform $g(s)$ to $g_1(u)$ using Eq. (5.7).
2. Compute the convolution kernel: $h_1(u-v) \leftarrow h(s,t)$ using Algorithm 5.1
3. Deconvolve: $g_1 = h_1 \circledast f_6$
4. Recover $\hat{f}_5(v)$ using $\hat{f}_5(v) = {\hat{f}_6(v)}\big/{f_4'(v)}$ (Eq. (5.9))
5. Invert domain transformation: $\hat{f}(f_4(v)) = \hat{f}_5(v)$ (Eq. (5.8))

Algorithm 5.3: Algorithm for applying GCT to solve the integral equation.

5.4. Verification

In this section, GCT is verified through numerical simulation experiment on a hypothetical image forming system in 1D. The image forming system is briefly described. Then, the transformation relations between the s,t domain and the new u,v domain are derived. That is, the expressions for functions f_1, f_2 and f_3, f_4 are derived.

5.4.1. Hypothetical image forming system

A schematic diagram of the 1D image forming system is shown in Fig. 5.1. In the figure, the real-world coordinates are to the left of the lens and image coordinates are on the right. The system is assumed to work within paraxial optics assumption. A 1D object placed parallel to the optical axis, as shown, forms a 1D image behind the lens. Such a system is governed by the image formation equation, Eq. (5.15).

$$\frac{1}{f_0} = \frac{1}{Z} + \frac{1}{z} \tag{5.15}$$

The intensity distribution shown in the figure is expressed by:

$$f(t) = c_0 + sin(\omega(t - a) + k_0) \tag{5.16}$$

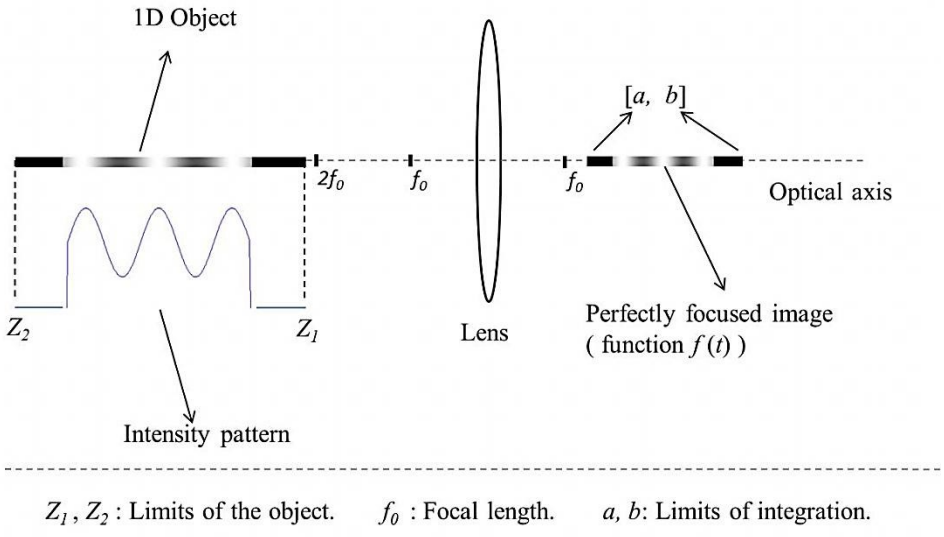


Figure 5.1: Showing the hypothetical image forming system

Defocus aberration is modeled in geometric optics by the cylindrical function. It is expressed by the following.

$$h(s,t) = \frac{1}{\pi(R(s,t)^2 + r_0^2)} \tag{5.17}$$

The radius R is given by,

$$R(s,t) = \frac{D}{2}\left(\frac{s}{t} - 1\right). \tag{5.18}$$

From Eq. (5.17) and (5.18), $h(s,t)$ is written as,

$$\begin{aligned} h(s,t) = h\left(\frac{s}{t}\right) &= \frac{1}{\pi\left(\left(\frac{D}{2}\frac{s}{t} - 1\right)^2 + r_0^2\right)} \\ \Rightarrow h\left(\ln\left(\frac{s}{t}\right)\right) &= \frac{1}{\pi\left(\left(\frac{D}{2}\ln\left(\frac{s}{t}\right) - 1\right)^2 + r_0^2\right)} \\ h(\ln(s) - \ln(t)) &= \frac{1}{\pi\left(\left(\frac{D}{2}(\ln(s) - \ln(t)) - 1\right)^2 + r_0^2\right)} \end{aligned} \tag{5.19}$$

Comparing the above expression with $h_1(f_1(s) - f_2(t))$, the forward transformation relations are obtained. Therefore, $h(s,t) = h\left(\frac{s}{t}\right)$ is in the required form to obtain the correct transformation relations according to GCT. The variables of transformation are,

$$u = f_1(s) = \ln(s)\,. \tag{5.20}$$

$$v = f_2(t) = \ln(t)\,. \tag{5.21}$$

From the above relations, the inverse transformations are readily derived. They are,

$$s = f_3(u) = \mathrm{e}^u\,. \tag{5.22}$$

$$t = f_4(t) = \mathrm{e}^t\,. \tag{5.23}$$

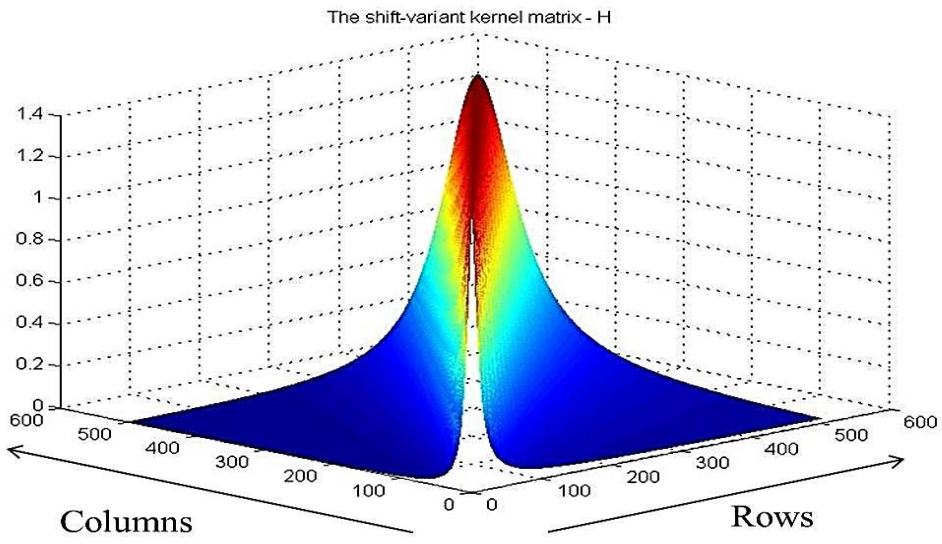
5.4.2. Numerical example

The transformation relations developed in 5.4.1 are used in the numerical example. The system parameters used are tabulated below.

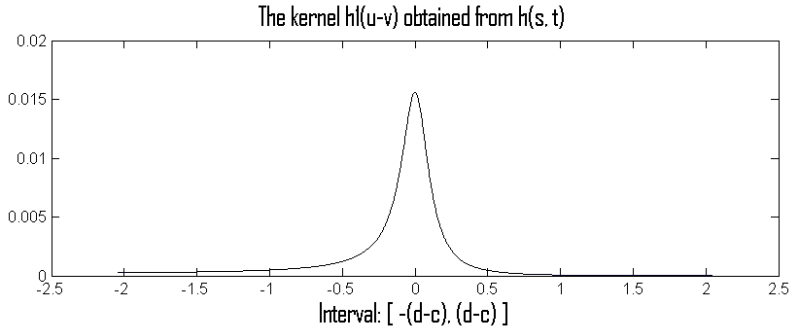
Parameter name	Value
Focal length - f_0	10 mm
F/#	2.8
Aperture – $D = \frac{f_0}{F/\#}$	3.5714
$[a, b]$	$[13, 100]$

Table 5.1: System parameters used for experiments in 1D.

The figure below shows the shift variant kernel $h(s,t)$ and the shift-invariant PSF $h_1(u-v)$ obtained through Algorithm 5.1 for the parameters in the Table 5.1.



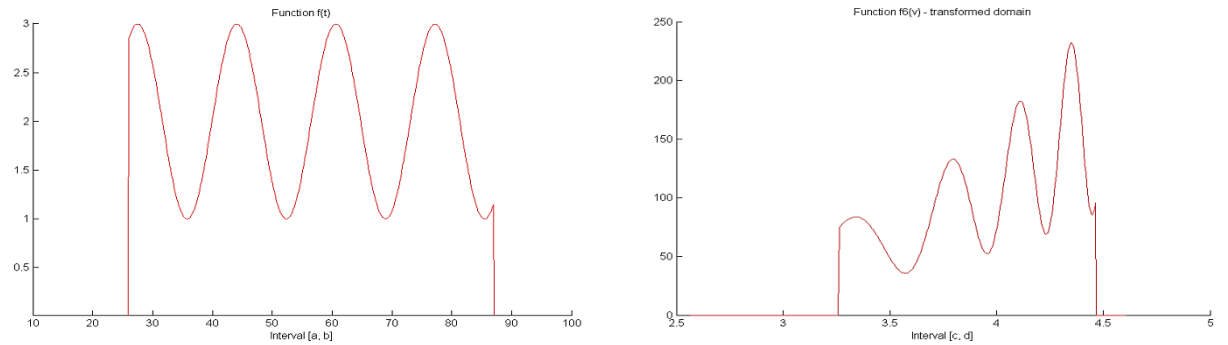
(a)



(b)

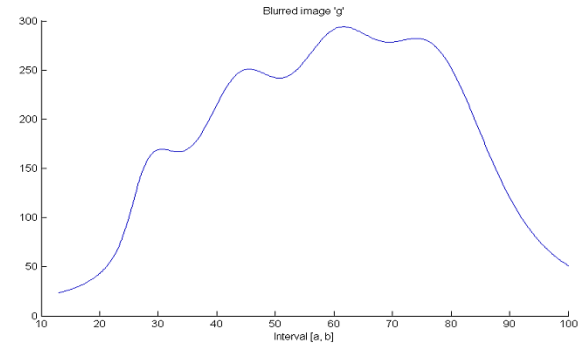
Figure 1.5.2: (a) Showing the shift-variant kernel matrix **H**. (b) Showing the shift-invariant PSF $h_1(u-v)$ obtained from **H**

The figure below shows different stages of a *forward* operation using GCT. Algorithm 5.2 was implemented in Matlab. The function $f(t)$ is of the form in Eq. (5.16). The number of sampling points was $N = 1000$.

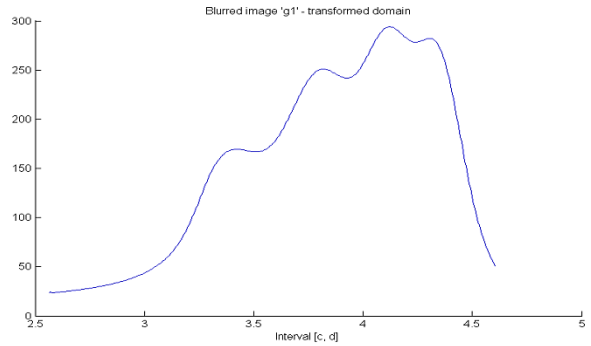


(a)

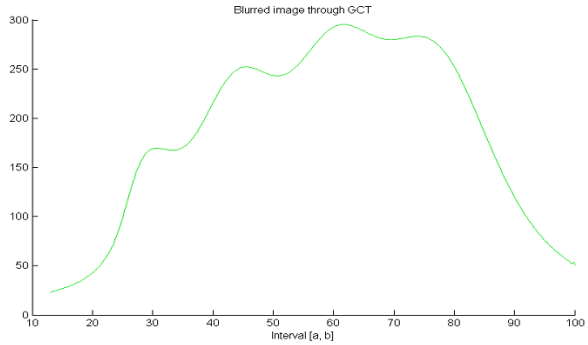
(b)



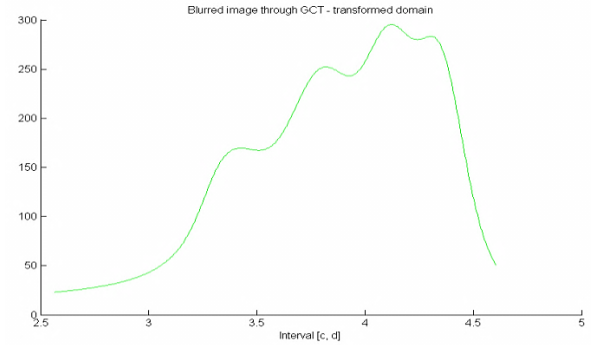
(c)



(d)



(e)



(f)

Figure 5.3: Showing forward operation through brute force computations and GCT. 3(a) and (b): Original function $f(t)$ and it in the transformed domain – $f_6(v)$. 3(c) and (d) Blurred function $g(s)$ in the original domain and the transformed domain respectively. 3(e) and (f): Blurred function computed through GCT in the original domain and the transformed domain respectively.

Figs. 5.3(b), (d) and (f) show the plots in the transformed domain. The last row of Fig. 5.3 show the results obtained from computations through GCT. From Figs. 5.3(c)

and (d) it is clear that the output of GCT matches very closely to the output of a brute force computation. The relative mean squared error between brute force computation and *forward* operations through GCT was found to be ~ 1e-5.

The *inverse* operation was also tested for the same example. The Deconvolution method used was the Landweber's algorithm [32]. The focused image estimate in the transformed domain $\widehat{f}_6(v)$ was obtained after 80 iterations. It was then transformed back to the original domain using spline interpolation. The *inverse* operation was implemented in Matlab and executed on an Intel Core i-3 2.4GHz CPU with 4GB of RAM. Total time of execution of Algorithm 5.3 was 0.4211 seconds. The total time of execution of direct inversion of the kernel matrix was 2.1971 seconds. That means GCT improved the computation time for *inverse* operation by a factor of five. The relative mean squared error between the original function and the estimate obtained through GCT was 0.0065 (or 0.65%). The relative mean squared error for direct inversion was 0.0012.

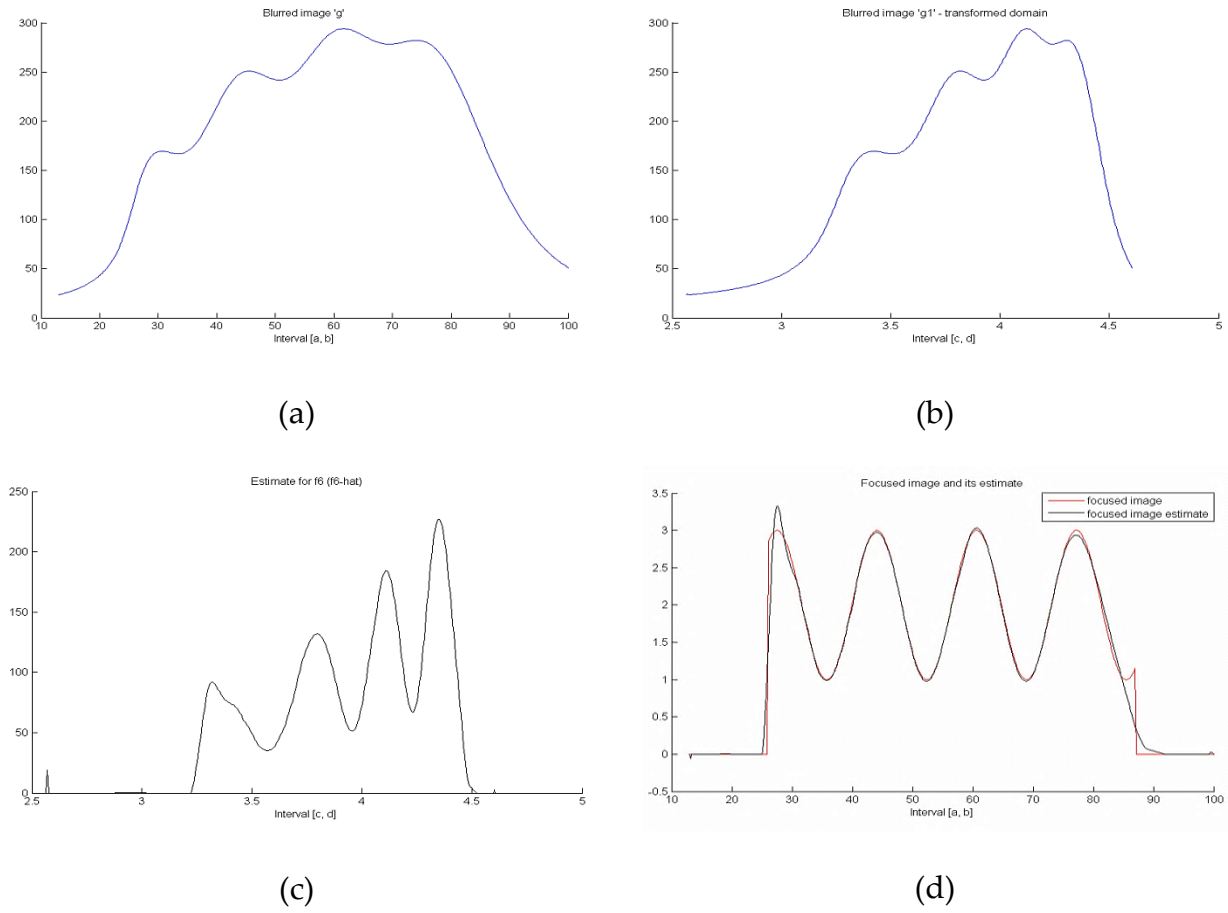


Figure 5.4: Results of *inverse* operation through GCT in 1D. 5.4(a) and 5.4(b) show the input to Algorithm 5.3; blurred image in the original and transformed domain. 5.4(c) shows the focused image estimate $\hat{f}_6(v)$. 5.4(d) shows the focused image estimate $\hat{f}$ (black) and f (red); relative MSE = 0.0065.

5.5. Multi-dimensional GCT

The following proof for multi-dimensional GCT was given by Dr. Satyaki Dutta (personal communication). Multi-dimensional GCT is a natural extension of the 1D GCT and provides the most general transformation of the Fredholm Integral Equation. The proof uses the change of variable theorem of multivariable calculus. The statement and proof are as follows.

Statement: Let

$$g(s_1, s_2, \dots s_n) = \int_R h(s_1, s_2, \dots s_n; t_1, t_2, \dots t_n)\, f(t_1, t_2, \dots t_n)\, dt_1\, dt_2 \dots dt_n \tag{5.24}$$

Where $R = [a_1, b_1] \times [a_2, b_2] \times \dots [a_n, b_n]$ and $a_i \leq s_i, t_i \leq b_i$ for $1 < i < n$. Assume that for $1 < i < n$ and $\alpha = s, t$, there exist functions φ_i^α defined on R with continuous derivatives such that $\Phi^\alpha \equiv (\varphi_1^\alpha, \varphi_2^\alpha, \dots \varphi_n^\alpha)$ defines change of variables, with inverse $\Psi^\alpha \equiv (\psi_1^\alpha, \psi_2^\alpha, \dots \psi_n^\alpha)$. For $1 < i < n$ define:

$$\varphi_i^s(s_1, s_2, \dots s_n) = \tilde{s}_i$$

$$\psi_i^s(\tilde{s}_1, \tilde{s}_2, \dots \tilde{s}_n) = s_i$$

$$\varphi_i^t(t_1, t_2, \dots t_n) = \tilde{t}_i$$

$$\psi_i^t(\tilde{t}_1, \tilde{t}_2, \dots \tilde{t}_n) = t_i$$

Then,

$$\begin{aligned} &g\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n)\right) \\ &= \int_R \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \\ &\quad f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\, J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{aligned} \tag{5.25}$$

If and only if

$$\begin{aligned} h\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n);\ \psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right) \\ = \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \end{aligned} \tag{5.26}$$

Where

$$J(\tilde{t}_1, \dots \tilde{t}_n) = \left(\frac{\partial \psi_i^t}{\partial \tilde{t}_j}\right) \tag{5.27}$$

is the Jacobian determinant.

Proof of if part:

By change of variables, we get:

$$g\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n)\right)$$
$$= \int_R \begin{matrix} h\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n);\ \psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right) \\ f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\ J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{matrix} \quad (5.28)$$

By assumption,

$$h\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n);\ \psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right) = \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \quad (5.29)$$

Therefore

$$g\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n)\right) = \int_R \begin{matrix} \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \\ f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\ J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{matrix} \quad (5.30)$$

Proof of only if part:

$$g\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n)\right)$$
$$= \int_R \begin{matrix} h\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n);\ \psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right) \\ f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\ J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{matrix}$$

$$= \int_R \begin{matrix} \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \\ f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\ J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{matrix}$$

Implies

$$\int_R \begin{matrix} h\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n);\ \psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right) \\ f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\ J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{matrix}$$

$$= \int_R \begin{matrix} \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \\ f\left(\psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right)\ J(\tilde{t}_1, \dots \tilde{t}_n) d\tilde{t}_1 \dots d\tilde{t}_n \end{matrix}$$

is true for any arbitrary function f . Therefore

$$h\left(\psi_1^s(\tilde{s}_1, \dots \tilde{s}_n), \psi_2^s(\tilde{s}_1, \dots \tilde{s}_n), \dots \psi_n^s(\tilde{s}_1, \dots \tilde{s}_n);\ \psi_1^t(\tilde{t}_1, \dots \tilde{t}_n), \psi_2^t(\tilde{t}_1, \dots \tilde{t}_n), \dots \psi_n^t(\tilde{t}_1, \dots \tilde{t}_n)\right) = \tilde{h}(\tilde{s}_1 - \tilde{t}_1, \tilde{s}_2 - \tilde{t}_2, \dots \tilde{s}_n - \tilde{t}_n) \quad (5.31)$$

Thus, the multi-dimensional GCT is proved.

5.6. Conclusions

The Generalized Convolution Theorem for transforming an LSV system to an LSI system under suitable conditions was proved in 1D. Based on the GCT, two computational algorithms were developed to perform the *forward* and *inverse* operations on an LSV system. An algorithm to compute the kernel in the transformed domain was also presented. The theorem proves that the necessary conditions for transforming an LSV system to an LSI system are (i) the kernel should be in a separable form as in Eq. (5.6), and (ii) the domain transformations f_1, f_2 are invertible and differentiable.

The correctness of the theorem was verified by a numerical example on a hypothetical 1D image-forming system. The improvement in computation time to obtain the focused image estimate from a blurred image was also highlighted. Although the GCT was applied to an imaging system, it is possible to apply it to any LSV system whose kernel satisfies the required conditions. In the next chapter, applications of GCT to 3D microscopy and 3D shape recovery are investigated.

6. Generalized Convolution Theorem: Applications

The Generalized Convolution Theorem (GCT) was introduced in the previous chapter. It was proved that with GCT, it is possible to transform some linear shift-variant (LSV) systems to linear shift-invariant (LSI) systems. Conditions to be satisfied by an LSV system's kernel for transformation to be possible were discussed in detail.

In 3D Microscopy, the objective is to recover a 3D image of an object through a set of defocused images acquired at different focal planes. It is possible to relate the observed 3D image, underlying focused image and 3D point spread functions as a linear system. In fact, such a system is shift-variant in the z (axial) variable. The focused image is then estimated through deconvolution. By applying GCT to 3D Microscopy, the computational complexity of shift-variant deconvolution is reduced to $O(N \log(N))$. This provides significant gains in computations compared to a piecewise constant shift-invariance (PCSI) approximation method. GCT based deconvolution models shift-variant system more accurately than shift-invariant (SI) approximation. Therefore it provides more accurate estimates compared to the standard SI and PCSI deconvolution techniques.

Another application of GCT is in the 3D shape recovery of surface objects. This is discussed with two simulation examples. It is demonstrated that GCT can recover the shape information quite accurately after restoration.

The rest of the chapter is organized as follows. The fundamentals of image formation in three dimensions are discussed in 6.1. A brief introduction to the literature in 3D microscopy is presented in Section 6.2, followed by a discussion on how to apply GCT to it. Details of computer implementation of GCT are discussed in Section 6.4. Sections 6.5 and 6.6 present results of simulation experiments of *forward* and *inverse* operations using GCT. Quantitative and qualitative comparisons with standard techniques and noise analysis are presented. In section 6.7 application of GCT towards shape recovery is briefly discussed with some examples.

6.1. Background

Consider a 3D object being imaged. It is made up of point light sources weighted by the intensity of light at that point. A *point light* source forms a perfectly focused *point image* at the location determined by lens maker's formula; Eq. (5.15) (Chapter 5).

A focused image surface (FIS) of a surface object is defined as the collection of its *image points* in the image space. Another way of (conceptually) defining FIS is: image of FIS is the object. A focused image volume (FIV) is the equivalent of FIS for volume objects.

It is impossible to either measure or observe an FIS (or FIV) with conventional imaging systems. However, it is possible to formulate a relationship between FIS (or FIV) and a sequence of images of the same scene, recorded by varying camera

parameters. In practice, it is common to move the image detector along the optical axis while keeping the other camera parameters fixed.

The following expression relates a sequence of images $g(x, y, z)$, 3D shift-variant kernel (PSF) $h(x, y, z, x_1, y_1, z_1)$ and FIS (or FIV) $i(x_1, y_1, z_1)$:

$$g(x, y, z) = \int_{xm}^{xn} \int_{ym}^{yn} \int_{a}^{b} i(x_1, y_1, z_1)\ h(x, y, z, x_1, y_1, z_1)\, dx_1\, dy_1 dz_1 \qquad (6.1)$$

The above equation is shift-variant in all three variables x, y and z. This way of modeling image formation is required for wave optics when the imaging system exhibits significant amounts of primary aberrations (Note: Spherical Aberration is shift-invariant in the XY plane).

A detailed discussion of modeling 3D images can be found here [71-74]. Since a planar object perpendicular to the optical axis is degraded by equal amount of blur at every pixel, the integral becomes shift-invariant in x and y variables and we obtain,

$$g(x, y, z) = \int_{xm}^{xn} \int_{ym}^{yn} \int_{a}^{b} i(x_1, y_1, z_1)\ h(x - x_1, y - y_1, z, z_1)\, dx_1\, dy_1 dz_1\ . \qquad (6.2)$$

The limits of integration $[a, b]$ are positions at which the first and the last 2D image are acquired by moving the detector. The 3D PSF of a point object is a collection of its 2D PSFs at different focal positions in the range $[a, b]$. The assumption that all parameters of imaging except the detector distance z remain fixed shapes the 3D PSF like an hourglass as shown in Fig. 6.1. In the figure, a 3D PSF of a point coming to focus at z_1 is shown. The symbol R is the *blur circle radius* of the 2D PSF and it is a function of image detector position z for a given point object. The aperture diameter is D. A derivation of the relation between these variables will follow. Note that the 3D PSF is

does not have local support domain; typically the domain occupies the entire interval [a, b] in the z variable.

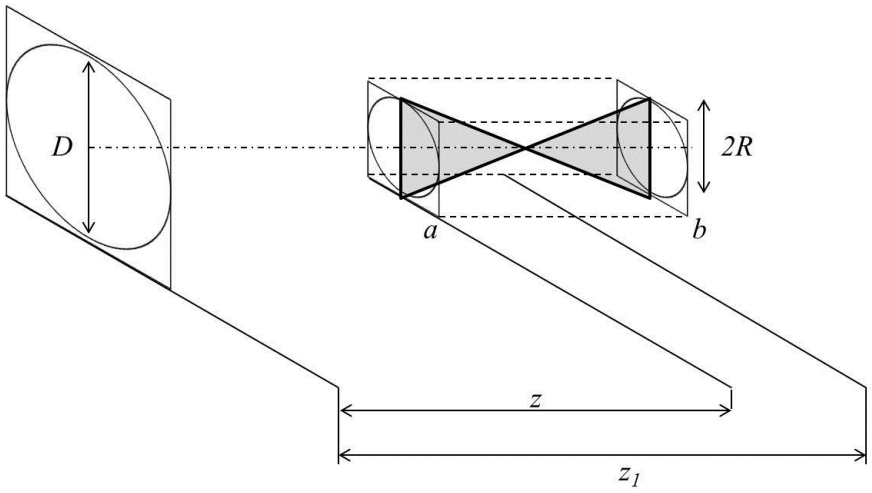


Figure 6.1: Schematic of 3D imaging system and 3D PSF

Fig. 6.2 below is a geometric diagram of image formation of a point object P under paraxial optics assumption. The point object P is located at a distance Z from the optic center O. The rays $\overrightarrow{PAp}$, $\overrightarrow{PBp}$ and $\overrightarrow{POp}$ meet at a distance z_1 from the optic center. The image detector is placed at a distance z from the optic center. The observed image p' is therefore defocused. The light from the point source is spread out into a circular region called the *blur circle* with radius R. Blur circle radius R is a function of variables z and z_1, such that the system in Eq. (6.2) is shift-variant. The relation between them is derived from similar triangles ΔABp and Δmnp as follows.

$$\frac{\overline{AB}}{z_1} = \frac{\overline{mn}}{z - z_1}$$

$$\Rightarrow \frac{D}{z_1} = \frac{2R}{z - z_1}$$

$$2R = \frac{D(z - z_1)}{z_1} \Rightarrow R = \frac{D}{2}\left(\frac{z}{z_1} - 1\right) \tag{6.3}$$

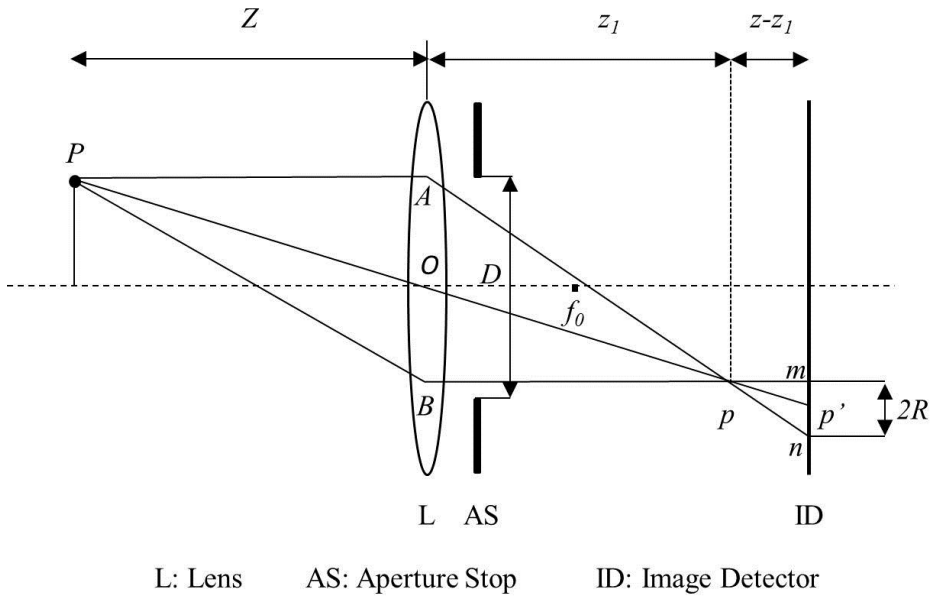


Figure 6.2: Schematic diagram of image formation under geometric optics

From Eq. (6.3), a cylindrical 3D PSF is given by,

$$h(x - x_1, y - y_1, z, z_1) = \begin{cases} \dfrac{1}{\pi R^2(z, z_1)} & for\ (x - x_1)^2 + (y - y_1)^2 < R^2 \\ 0 & otherwise. \end{cases} \tag{6.4}$$

Noting that the blur circle radius is proportional to the standard deviation σ of a Gaussian PSF model, the expression for it is written as follows.

$$h(x - x_1, y - y_1, z, z_1) = \frac{1}{2\pi\,\sigma^2(z, z_1)} \exp\left(\frac{(x - x_1)^2 + (y_1 - y)^2}{2\sigma^2(z, z_1)}\right) \tag{6.5}$$

6.2. Introduction to 3D microscopy

Imaging of biological microscopic objects is performed by a high magnification microscope to construct its 3D model. One way to do that is by using a confocal microscope [75]. In a Confocal Scanning Microscope a pin-hole is placed just in front of the detector. Then light from different axial sections of the object is scanned by axially moving the object. The pin-hole filters the light coming from planes that are not in focus. Hence, a direct 3D image of microscope object is constructed. Some disadvantages of confocal microscopy are: (i) the acquired images contain low light hence more noise, (ii) the process of acquisition is slow which limits their imaging capability to non-live cells/tissues [59].

An alternative to confocal microscopy is provided by Widefield Microscope Imaging. Widefield microscopes collect light from the section that is in focus (or the focal plane) as well as light coming from the sections behind, and in front of, the focal plane. This essentially degrades the image of the focal plane by blurring it with light from outside the plane. The blurred image is then processed by a computer to recover the underlying 3D focused image of the object. From here on, 3D deconvolution refers to widefield microscopy.

Early investigations in 3D deconvolution were carried out by Castleman, and Agard and Sedat [4, 76-78]. Castleman derived three-dimensional convolution approximation to microscope image formation. The blurred image of each focal plane was processed by considering only a few neighboring planes [76]. Such an algorithm is highly efficient in terms of computational requirements however, do not produce accurate solutions. Agard et al processed the 3D data by deconvolving the blurred image in an iterative fashion. Erhardt et al approached the problem of deconvolution through linear system

theory [79]. They used inverse filtering in the frequency domain to deconvolve the entire 3D blurred image. Using the linear system framework, many 2D deconvolution algorithms have been extended to deconvolution microscopy [59, 80-82]. Some of the standard methods include Lucy-Richardson deconvolution [60], Landweber's method [83, 84], maximum-likelihood techniques [84] and constrained iterative algorithms [85]. A survey of different deconvolution techniques, including their performance evaluations can be found here [86]. All the above methods assumed that the PSF is known. Some blind deconvolution techniques have been proposed to handle cases where PSF of the imaging system is not adequately modeled theoretically [15, 59].

Gibson and Lanni derived an analytical expression to accurately model microscope PSFs [72] and made the case for axial shift-variance of 3D imaging systems. In general, for thick specimens the axial shift-variance becomes a problem. Kam et al proposed an adaptive iterative algorithm that considers shift-variance of 3D PSFs [87, 88]. Hanser et al made phase measurements to recover the shift-variant PSF and performed a blockwise restoration using the Lucy-Richardson algorithm [89]. Blockwise restoration involves deconvolving the sub-volume data with different PSFs and assembling them at the final step to construct the complete 3D object. The sub-volumes are small enough so that the change in PSF is minimal. Such methods produce discontinuities between sub-volumes of the constructed 3D object. Preza et al developed methods that deconvolve non-overlapping strata (or blocks) independently and then mitigate the problem of discontinuity by performing interpolation of PSFs between different strata [90, 91].

At present, a 3D shift-variant blurred image is deconvolved by processing a number of small shift-invariant blocks, say k. Due to large support of the PSF, complete data has to be processed by k shift-invariant PSFs (one per block). Then from each restored images, only the block of restored data that corresponds to its PSF is retained. Such an

approach is computationally expensive both in terms of time and memory storage. With the help of GCT, shift-variant imaging system is transformed into a shift-invariant system without losing much accuracy. Then, the 3D blurred image is restored in the transformed domain through simple iterative deconvolution. This improves the computation time by a factor of k compared to a blockwise shift-variant deconvolution. The transformations require interpolation only along the axial direction hence do not add much overhead in the processing of 3D blurred images.

In the following section, the transformation relations for a 3D imaging system are derived under geometric optics assumption.

6.3. Derivation of transformation relations

Eq. (6.2) is the shift-variant image formation equation in 3D. It is shift-variant in only the z variables. Therefore, 1D GCT is applied to z and z_1 variables to transform Eq. (6.2) to a convolution integral in 3D. From Eqns. (6.3), (6.4) and (6.5), the 3D PSF h can be written in the following form.

$$h(x - x_1, y - y_1, z, z_1) = h\left(x - x_1, y - y_1, \frac{z}{z_1}\right)$$

$$\Rightarrow h\left(x - x_1, y - y_1, \frac{z}{z_1}\right) = h_1\left(x - x_1, y - y_1, \log\left(\frac{z}{z_1}\right)\right)$$

$$= h_1\big(x - x_1, y - y_1, \log(z) - \log(z_1)\big) = h_1(x - x_1, y - y_1, z_4 - z_5) \tag{6.6}$$

From the above equations, the transformation relations are obtained:

$$f_1(z) = z_4 = \log(z) \tag{6.7}$$

$$f_2(z_1) = z_5 = \log(z_1) \tag{6.8}$$

The inverses transformations are as follows:

$$f_3(z_4) = z = e^{z_4} \tag{6.9}$$

$$f_4(z_5) = z_1 = e^{z_5} \tag{6.10}$$

$$f'_4(z_5) = dz_1 = e^{z_5} dz_5 \tag{6.11}$$

After transformations, the blurred image g and the focused image i become:

$$g_1(x, y, z_4) = g\big(x, y, f_3(z_4)\big) \equiv g(x, y, z), \tag{6.12}$$

$$i_1(x_1, y_1, z_5) = i\big(x_1, y_1, f_4(z_5)\big) \equiv i(x_1, y_1, z_1)\,. \tag{6.13}$$

Changing the variables z and z_1 in Eq. (6.2) according to the relations in Eqns. (6.7) and (6.8), we get:

$$g\big(x, y, f_3(z_4)\big) =$$

$$\int_{xm}^{xn}\int_{ym}^{yn}\int_{a}^{b} i\big(x_1, y_1, f_4(z_5)\big)\; h\big(x - x_1, y - y_1, f_3(z_4), f_4(z_5)\big)\, dx_1\, dy_1 df_4(z_5) \tag{6.14}$$

From Eqns. (6.6), (6.11), (6.12) and (6.13), the above equation is rewritten as,

$$g_1(x, y, z_4) =$$

$$\int_{xm}^{xn}\int_{ym}^{yn}\int_{c=f_2(a)}^{d=f_2(b)} i_1(x_1, y_1, z_5)\; h_1(x - x_1, y - y_1, \mathrm{z}_4 - \mathrm{z}_5)\; (e^{z_5})\; dx_1\, dy_1 \mathrm{dz}_5 \tag{6.15}$$

The scaling factor (e^{z_5}) resulting from the change of variables is taken inside the function i_1 and the above equation is rewritten as follows.

$$g_1(x, y, z_4) = \int_{xm}^{xn} \int_{ym}^{yn} \int_{c}^{d} i_2(x_1, y_1, z_5)\ h_1(x - x_1, y - y_1, \mathrm{z}_4 - \mathrm{z}_5)\, dx_1\, dy_1 d\mathrm{z}_5 \qquad (6.16)$$

Where,

$$i_2(x_1, y_1, z_5) = i_1(x_1, y_1, z_5)\ e^{z_5}, \quad c \le z_5 \le d\,. \qquad (6.17)$$

6.4. Computer implementation

Eq. (6.16) is in the continuous domain; however, the images observed are discrete. The imaging model in Eq. (6.16) is represented by 3D convolution operation as follows.

$$g_1 = i_2 \circledast h_1 \qquad (6.18)$$

The above equation is an approximation to the *forward operation* in Eq. (6.16). The 3D PSF does not have local support along z. To prevent coarse approximation, a large aperture has to be used so that the PSF values become negligible within a few planes from the Gaussian image point. The algorithm for computing the *forward operation* using GCT for a given focused image i and PSF h is as follows.

Forward operation using GCT (3D)

1. Interpolate $i_2(x_1, y_1, z_5)$ from $i(x_1, y_1, z_1)$ using Eqns. (6.10), (6.13) and (6.17);
2. Compute $h_1(x - x_1, y - y_1, \mathrm{z}_4 - \mathrm{z}_5)$ using Eq. (6.6); assume $z_5 = \frac{(c+d)}{2}$; $c \le z_4 \le d$
3. Compute $G_1(u, v, w) = I_2(u, v, w).H_1(u, v, w)$ using FFT;
4. Using inverse FFT, compute $g_1(x, y, z_4) \overset{DFT}{\Longleftrightarrow} G_1(u, v, w)$;
5. Transform $g_1(x, y, z_4)$ using: $g(x, y, z) = g_1(x, y, f_3(z_4))$; Eq. (6.12)

Algorithm 6.1: To compute the 3D blurred image using GCT given a focused image and the shift-variant kernels.

The procedure for computing the focused image estimate from a given blurred image with known PSF is similar to the algorithm discussed in the previous chapter. For the sake of completeness, the algorithm to compute the *inverse operation* using GCT is presented below.

Inverse operation using GCT (3D)

1. Transform $g(x,y,z)$ to $g_1(x,y,z_4)$ using $g_1(x,y,z_4) = g\big(x,y,f_3(z_4)\big)$
2. Compute $h_1(x-x_1, y-y_1, \mathrm{z}_4 - \mathrm{z}_5)$ using Eq. (1.6); assume $z_4 = \frac{(c+d)}{2}$; $c \le z_5 \le d$
3. Deconvolve: $g_1 = i_2 \circledast h_1$
5. Recover $i_1(x_1,y_1,z_5)$ using $i_1(x_1,y_1,z_5)e^{z_5} = i_2(x_1,y_1,z_5)$
6. Invert domain transformation: $i(x_1,y_1,z_1) = i_1\big(x_1,y_1,f_4(z_5)\big)$ through interpolation

Algorithm 6.2: To estimate the focused image using GCT given blurred image and shift-variant kernels

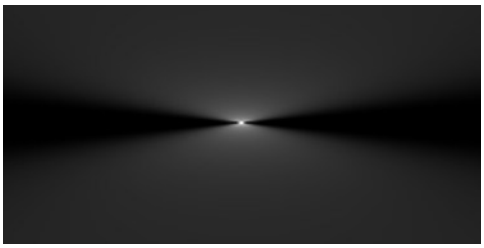
Note that step 3 in the above algorithm suggests that any deconvolution method could be used. The behavior of deconvolution methods depend on the shape and structural properties of the underlying object. In practice, iterative methods like Lucy-Richardson or Landweber iteration are used. In the presence of significant amount of noise some constrained regularization is performed for estimating the focused image.

6.5. Forward operation simulation

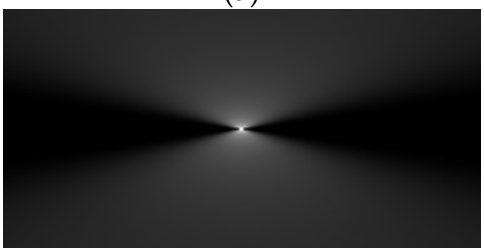
In this section, the blurred image is computed using GCT based *forward operation* algorithm. It is then compared with shift-invariant and piecewise constant shift-invariant approximations. The blurred image computed using a brute force implementation of Eq. (6.2) is the reference against which the accuracy of other blurred images is measured. The metric used for measuring the accuracy is the relative mean squared error:

$$ReMSE = \frac{\sum(g - \hat{g})^2}{\sum(g)^2} \tag{6.19}$$

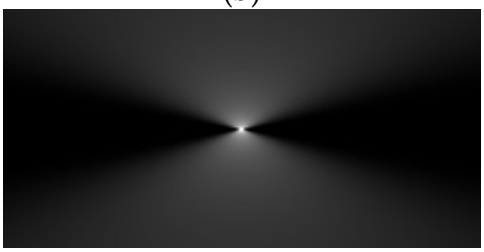
The figure below shows shift-variant PSFs.



(a)



(b)



(c)

Figure 6.3: Shows cross section of 3D Gaussian PSFs at different points along the axis; (a) PSF at $z = a + 0.15\,(\mathrm{b} - \mathrm{a})$; (b) PSF at $z = a + 0.5(b - a)$; (c) PSF at $z = a + 0.8(b - a)$. Camera parameters, $f_0 = 10\,mm$, $D = 3.5714\,mm$ and interval $[a, b] = [75, 250]$

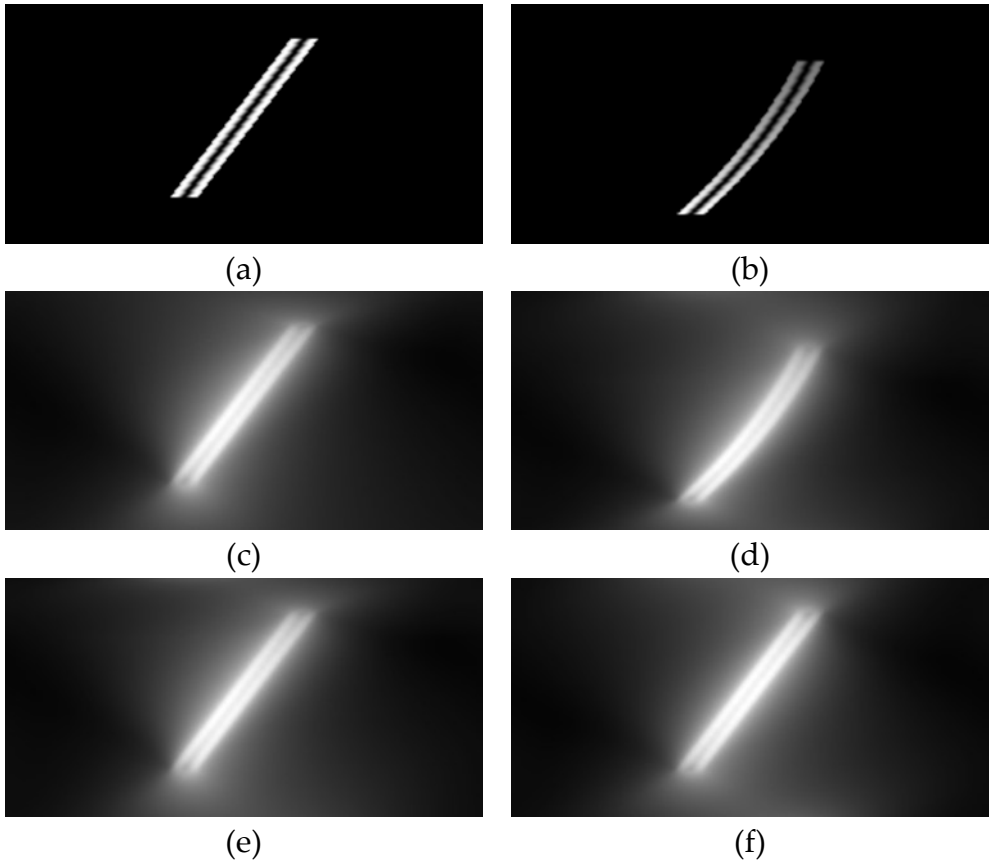


Figure 6.4: Shows focused image and blurred images obtained from different techniques; (a) Focused image; (b) Focused image after transformation; (c) Actual blurred image; (d) GCT blurred image in the transformed domain; (e) GCT forward operation blurred image ReMSE with (c) = 0.0031; (f) PCSI blurred image ReMSE with (c) = 0.0053 . The images are cross sections in the YZ plane containing the midpoint in the x variable.

Figure 6.3 shows the variation of PSFs along the z axis. The change in the shape of the PSF is clearly visible. The imaging parameter $[a, b] = [75, 250]$ corresponds to an object located between 10.42 to 11.54 mm from the lens. The degree of change in the PSF increases for thicker objects. Fig. 6.4(a) shows the cross section of the object – Pipes. The blurred images of the object computed through GCT and PCSI method match very closely to the blurred image computed by brute force. However, GCT does slightly better in terms of accuracy. It was observed that ReMSE between SI approximation and

the brute force blurred image was 0.0173. GCT based method was about three times faster than the PCSI method with three shift-invariant blocks.

6.6. 3D deconvolution using GCT

The GCT based deconvolution algorithm discussed in the previous section is analyzed from three perspectives. First, some simulations are carried out by varying the size of the interval $[a, b]$. Increasing the size makes the system more shift-variant. The behavior of GCT based deconvolution is analyzed and focused image estimates are compared with results from SI and PCSI approximations. Next experiment considers test objects of different sizes to compare the computation times of GCT, SI and PCSI deconvolution. The final experiment discusses restoration of noisy blurred 3D images. The Lucy-Richardson algorithm [38, 39] was used in GCT, SI and PCSI deconvolution in all experiments. For all the experiments the following imaging parameters were kept fixed.

Parameter name	Value
Focal length f_0	10 mm
F/#	2.8
Aperture $D = \frac{f_0}{F/\#}$	3.5714

Table 6.1: System parameter values that were kept fixed are listed above.

6.6.1. Experiment 1 – varying $[a, b]$

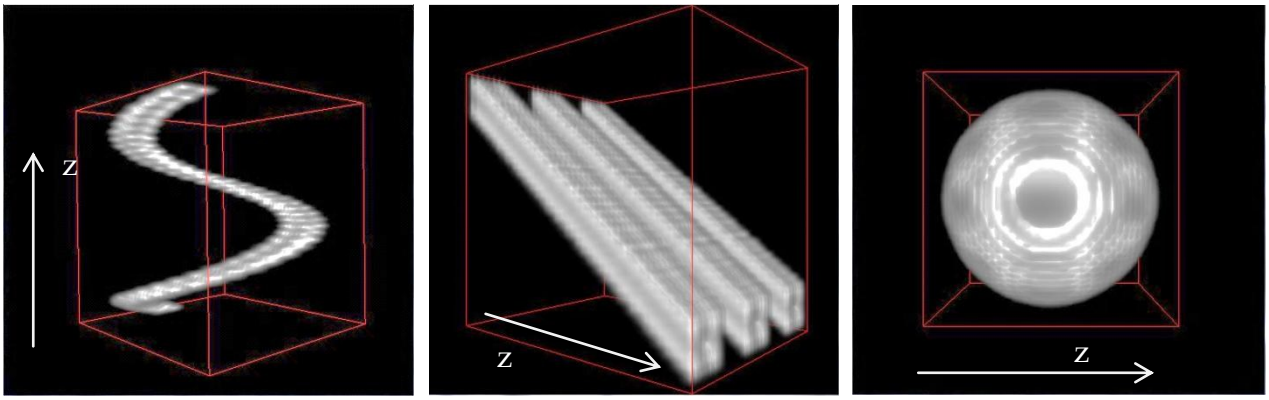
The following table presents results of deconvolution with three different intervals. The size of the interval is changed by increasing the value of b. For a lens of focal length 10 mm, image space location of 20 mm corresponds to object located on a plane

at a distance of 20 mm from the lens. Similarly, the location 60 mm, 100 mm and 170 mm correspond to object at 12 mm, 11.11 mm and 10.625 mm respectively. All the intervals correspond to objects that are less than 1 cm long.

The readings for different methods are the relative mean squared error between the estimate and the actual object. From the table it is evident that GCT improves accuracy of solution for shift-variant deconvolution. To be more precise, GCT is useful when the system is significantly shift-variant. Earlier, it was said that due to full support of the PSF, discretization of the imaging interval inherits some errors. For the purpose of this discussion, let the error be called Type A. Additional error is present in the shift-invariant model due to approximation. Let this error be called Type B. The GCT based deconvolution is effective in reducing Type B errors, which is the error resulting from shift-invariance approximation of a shift-variant system. GCT models the shift-variant system accurately, thus the relative mean squared error is expected to be less for a shift-variant system.

Object type	Method	[a, b] = [20, 60]	Iterations	[a, b] = [20, 100]	Iterations	[a, b] = [20, 170]	Iterations
Pipes	GCT	0.0917	60	0.0426	200	0.0366	200
	SI	0.1641	60	0.1609	60	0.1695	60
	PCSI	0.1121	60	0.1316	60	0.1391	60
Helix	GCT	0.166	140	0.0915	150	0.0995	200
	SI	0.24	80	0.2325	70	0.2285	70
	PCSI	0.1978	80	0.1678	70	0.1626	70
Sine	GCT	0.0654	45	0.0387	120	0.0314	170
	SI	0.1102	50	0.1033	45	0.1086	40
	PCSI	0.12	50	0.0875	45	0.0923	40

Table 6.2: Shows minimum relative mean squared error and the number of iterations of Lucy-Richardson deconvolution required to achieve it. The three rows correspond to three objects: Pipes, Helix and Sine. The columns highlight different sizes of the interval $[a, b]$ that were considered for this set of experiments.



(a) (b) (c)

Figure 6.5: Showing volume rendering of (a) Helix object, (b) Pipes object and (c) Sine object. (Images generated by ImageJ software.)

The figure above shows three example objects Helix, Pipes and Sine that were the input in simulation experiments. Fig. 6.5(a) shows an object in which the intensity in the volume is a pipe in the shape of a coil. This is referred to by the name Helix in the discussions. Fig. 6.5(b) shows three hollow pipes; it is called by the name Pipes in the discussions. Fig. 6.5(c) shows a sphere in which the light intensity is varying as a sinusoidal wave in all three directions. This object is simply called Sine.

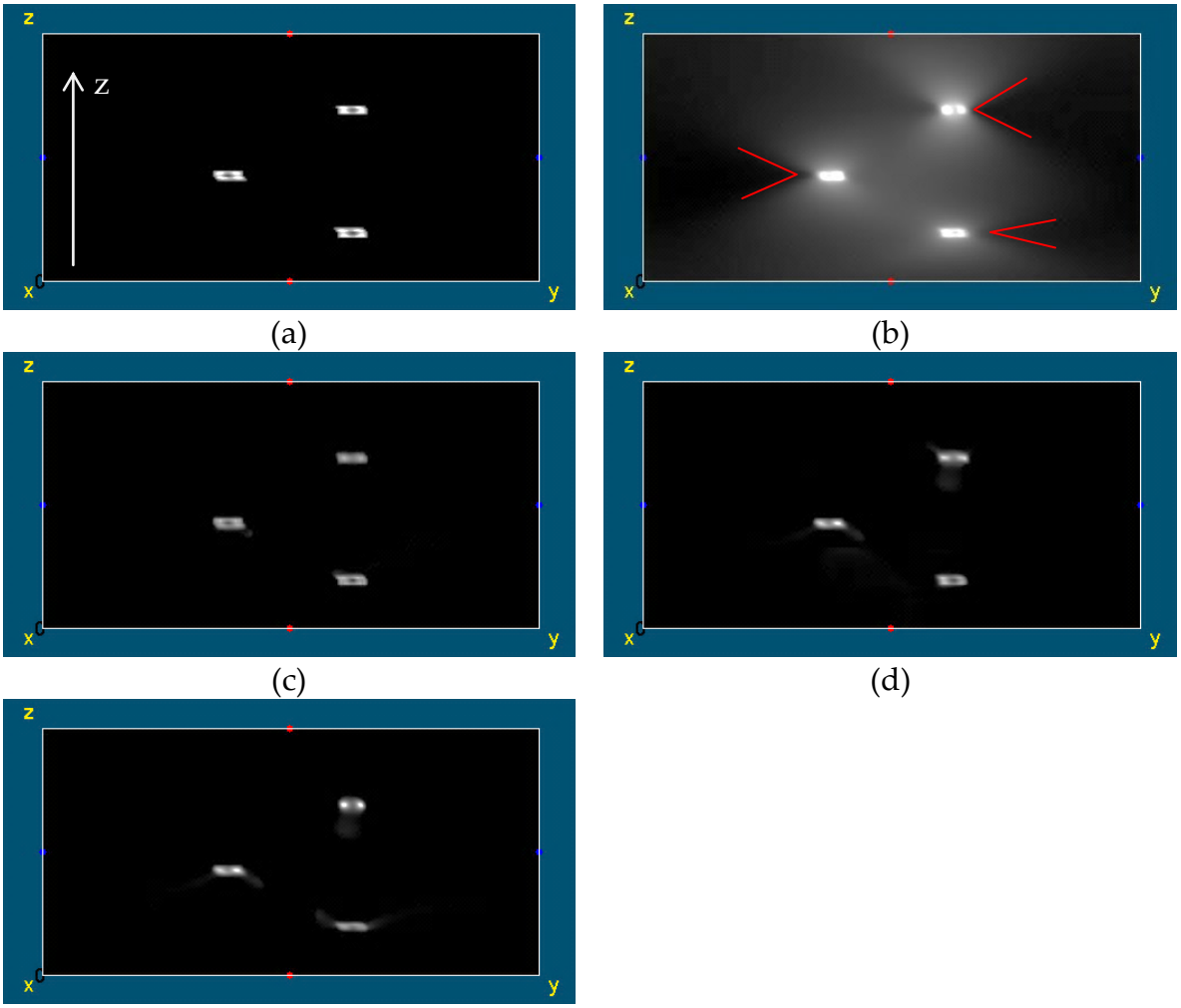
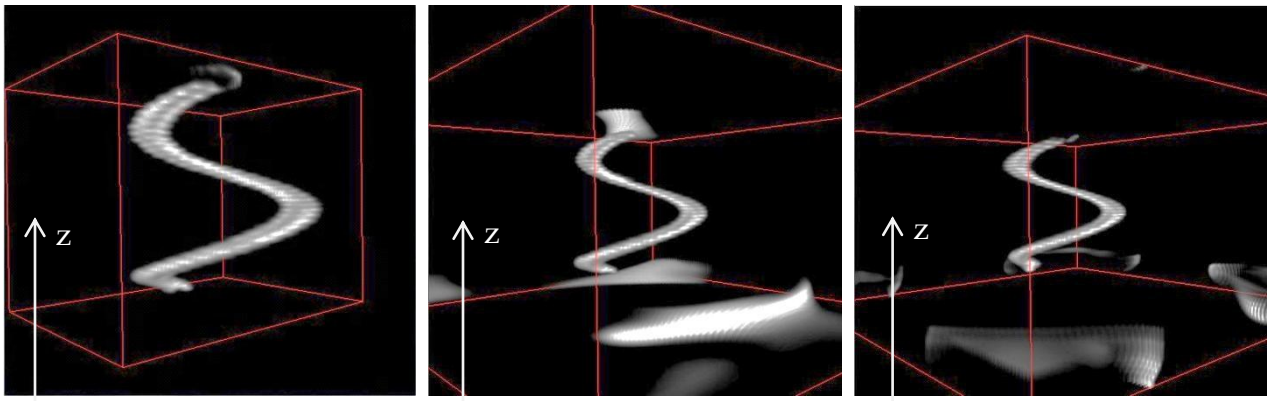


Figure 6.6: (a) Helix volume object sliced at the center in YZ plane. (b) Blurred image; the markers indicate change in the shape of the PSF. Restoration by: (c) GCT, (d) PCSI and (e) SI deconvolution. The markers indicate the axes. The limits of integration for this example are $[a, b] = [20, 100]$. Refer Table 6.2 for relative mean squared errors.

Figure 6.6 presents a case of deconvolution of the Helix object. The images are sliced through the center of the object in the YZ plane. In fig. 6.6(b), the red colored markers highlight the shift-variant nature of the system. Fig. 6.7 shows the 3D views of the reconstructed objects. It is apparent from the figure that the reconstruction from PCSI

and SI deconvolution introduce some structural artifacts that are absent in the result from GCT deconvolution.



(a) (b) (c)

Figure 6.7: 3D view of recovered Helix object, (a) GCT, (b) PCSI and (c) SI deconvolution

In Figs. 6.8 and 6.9, the restoration of blurred Pipes object and 3D views of the reconstructed objects are presented respectively. Figure 6.8(d) shows results from PCSI deconvolution that has some blocking artifacts. They are visible in figure 6.9(b) as well. The original Pipes object has uniform intensity pattern along z direction. The restoration from SI deconvolution near $z = a$ has visibly less intensity. This is due to shift-invariance approximation.

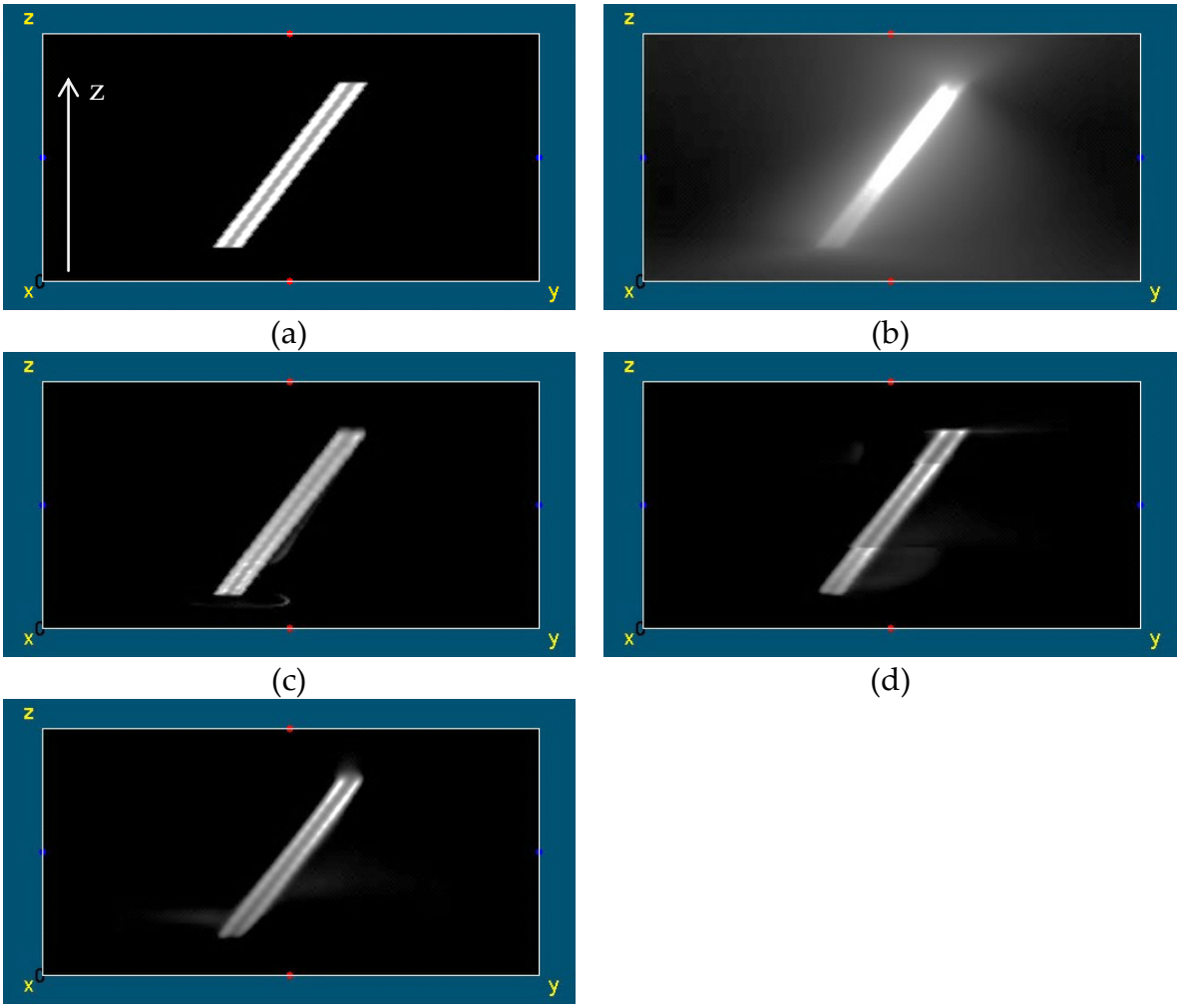
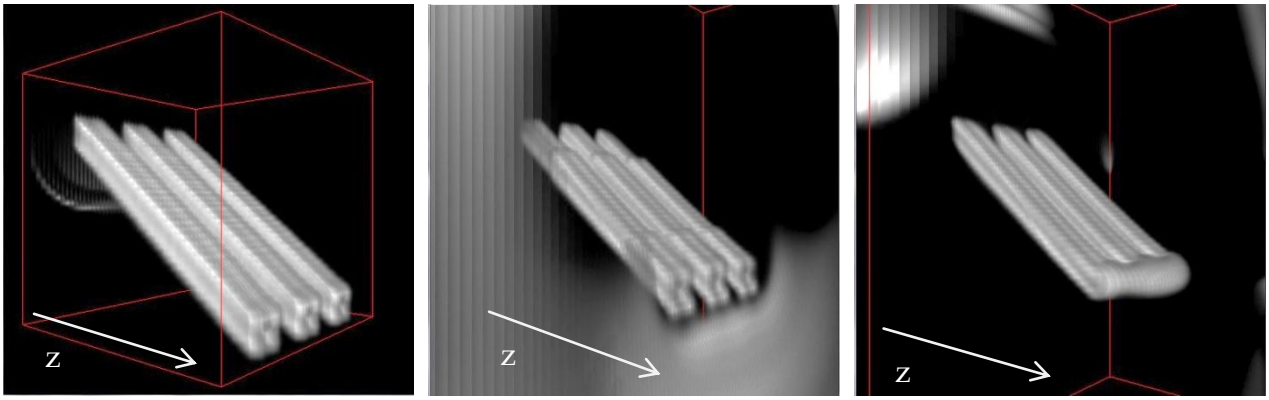


Figure 6.8: (a) Pipes volume object sliced at its center and along the YZ plane. (b) Blurred image. Restoration by: (c) GCT, (d) Piecewise constant shift-invariance (PCSI) and (e) Shift-invariance approximation. Note that PCSI method introduces blocking like artifacts after restoration. The markers indicate the axes. The limits of integration for this example are $[a, b] = [20, 170]$. Refer Table 6.2 for relative mean squared errors.



(a) (b) (c)

Figure 6.9: 3D view of recovered Pipes object, (a) GCT, (b) PCSI and (c) SI deconvolution

Figure 6.10 and 6.11 show original, blurred and deconvolved Sine object. Blocking artifacts from PCSI are visible in Fig. 6.10(d) and 6.11(b). The reconstruction from GCT deconvolution appears stretched. This type of behavior has been attributed to Fourier based techniques in the literature [92]. However, GCT based deconvolution does minimum deformation to the spherical structure as is evident from the 3D views in Fig. 6.11.

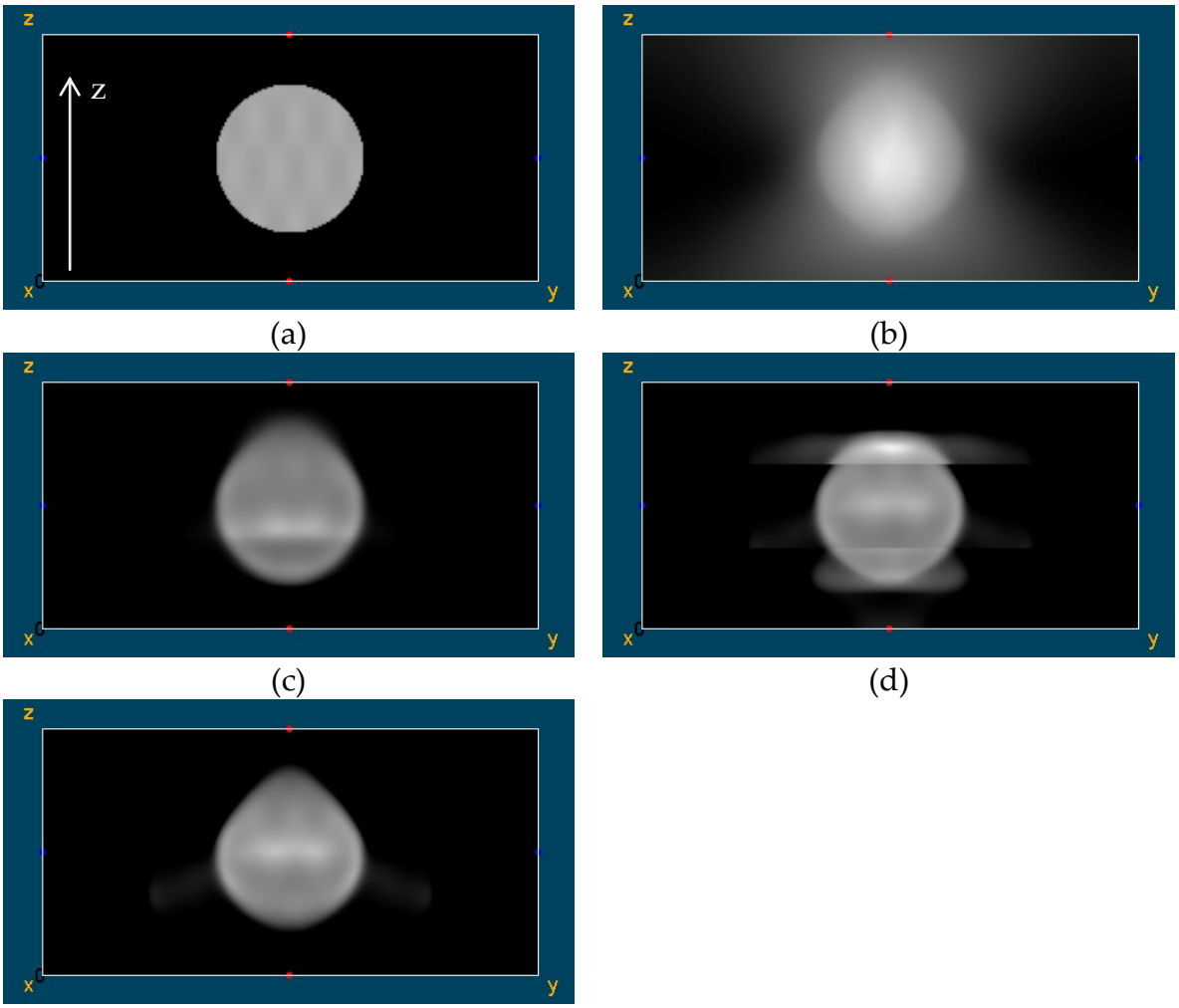
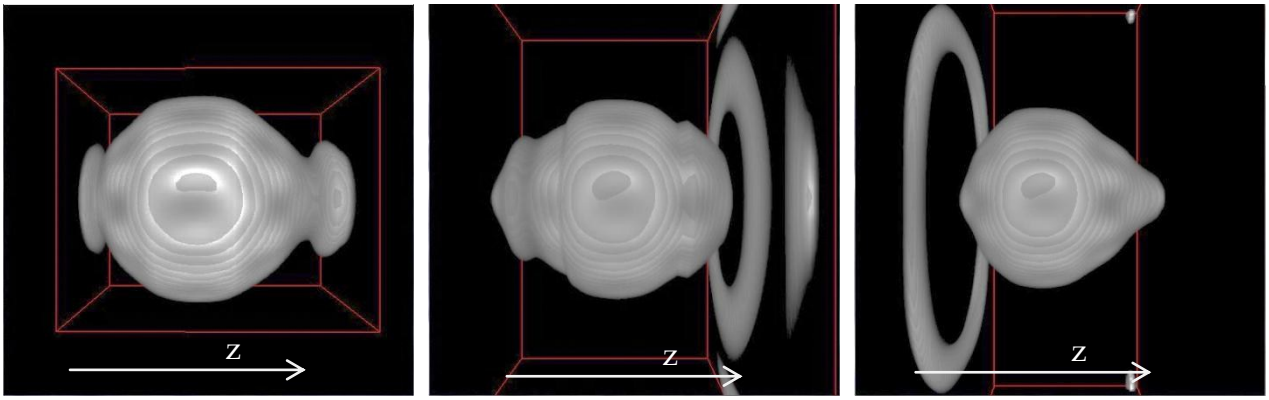


Figure 6.10: (a) Pipes volume object sliced at the center in yz plane. (b) Blurred image. Restoration by: (c) GCT, (d) Piecewise constant shift-invariance (PCSI) and (e) Shift-invariance approximation. Note that PCSI method introduces blocking artifacts after restoration. The markers indicate the axes. The limits of integration for this example are $[a, b] = [20, 60]$. Refer Table 6.2 for relative mean squared errors.



(a) (b) (c)

Figure 6.11: 3D view of reconstructed Sine object, (a) GCT, (b) PCSI and (c) SI deconvolution

6.6.2. Experiment 2 - Computation time measurement

For computation time analysis the Helix object was created at three resolutions, $128 \times 128 \times 63$, $192 \times 192 \times 95$ and $256 \times 256 \times 127$ voxels. The imaging parameters in Table 6.1 were used along with the interval of integration being $[a, b] = [20, 100]$. Lucy-Richardson deconvolution algorithm was used to obtain the 3D focused image estimate. The number of iterations for which SI deconvolution converged was used as the limit for GCT and PCSI deconvolution methods. Running the three algorithms for same number of iterations helps to compare the computation time in a straightforward way. The algorithms were executed on Intel Core i3 2.54 GHz processor with 4GB memory. The implementation was in Matlab and the code was not optimized to reduce computation time.

The table below shows the results of the time measurement experiment. It is also worth noting that GCT deconvolution method provided highest accuracy for the same number of iterations. Thus, GCT deconvolution converges faster towards the solution than PCSI or SI deconvolution methods. A k blocks PCSI algorithm deconvolves the

complete data k times. Therefore, both SI and GCT deconvolution are roughly a factor of k faster than PCSI deconvolution. The value of k for PCSI in the table was 3. Therefore in the table it is seen that GCT and PCSI algorithms are roughly three times faster. However, GCT deconvolution produced more accurate results.

Compared deconvolution in the original domain using SI approximation, the GCT deconvolution involves interpolating the data in the transformed domain. Such interpolations are computed twice. First interpolation is for computing the blurred image in the transformed domain. The second interpolation is for transforming the focused image estimate back to the original domain. Since only the z axis is transformed under GCT, interpolation is required for only one dimension of the three. Using this knowledge a simple linear interpolation scheme was implemented that is fast and reasonably accurate. The measured computation time includes the time taken for performing both the interpolations. It is evident from the table that the overhead of computing the interpolations is very less compared to the time taken for estimating the solutions.

Size	Iterations	GCT		PCSI		SI	
		Time	ReMSE	Time	ReMSE	Time	ReMSE
128 × 128 × 63	70	25.957s	0.0442	78.1513s	0.0846	25.9522s	0.0921
192 × 192 × 95	95	125.5615s	0.0565	373.7095s	0.0961	125.072s	0.1374
256 × 256 × 127	75	238.2996s	0.1513	715.3748s	0.1636	238.576s	0.2322

Table 6.3: Shows computation time measurements for GCT, PCSI and SI deconvolution for Helix object tabulated row-wise at different resolutions.

6.6.3. Experiment 3 - Deconvolution of noisy data

The final set of simulation experiments was on noisy data. Gaussian noise was added at three different power levels; 0.1%, 1% and 10% of signal power. The amount of noise added is indicated by signal-to-noise ratio (SNR) in the discussion. Lucy-Richardson algorithm was implemented to deconvolve noisy blurred images. To provide some robustness against noise, each image in the stack was smoothed by a 5×5 Gaussian filter of $\sigma = 1.5$ pixels. Such a smoothing operation has been proved to provide reasonably accurate results [75]. The imaging parameters used are listed in Table 6.1. The interval of integration was fixed to be $[a, b] = [20, 100]$. The following table compares the results of deconvolution of noise-free and noisy image.

Object type:	Method	Noise free	Iterations	30dB SNR	Iterations	20dB SNR	Iterations	10dB SNR	Iterations
Pipe	GCT	0.0426	200	0.0619	100	0.0626	100	0.0684	100
	SI	0.1609	60	0.1677	80	0.1679	80	0.169	80
	PCSI	0.1316	60	0.1284	80	0.1286	80	0.1340	80
Helix	GCT	0.0915	150	0.1364	150	0.137	150	0.1391	150
	SI	0.2325	70	0.2585	70	0.2585	70	0.259	80
	PCSI	0.1678	70	0.1983	70	0.198	70	0.1906	80
Sine	GCT	0.0387	120	0.0441	110	0.053	100	0.0637	100
	SI	0.1008	45	0.1045	45	0.1081	45	0.1115	55
	PCSI	0.0875	45	0.0901	45	0.0918	45	0.1041	55

Table 6.4: ReMSE for GCT, PCSI and SI deconvolution at 30 dB, 20 dB and 10 dB SNR are tabulated. The rows list the errors for different objects, and the amount of noise varies along columns. The columns labeled "Iterations" list the number of iterations of Lucy-Richardson required to obtain listed accuracies/ReMSE. The input for all the methods were smoothed versions of noisy blurred image.

From the table, it is noted that GCT deconvolution loses some accuracy while restoring noisy data. It was observed that the scaling operation, which is part of the final interpolation step, could amplify noise and reduce the accuracy of the solution in

the original domain. Therefore, smoothing of noisy input data is necessary. For both PCSI and SI deconvolution scheme, the accuracy of the solutions did not change much with added noise, as compared to the noise free case.

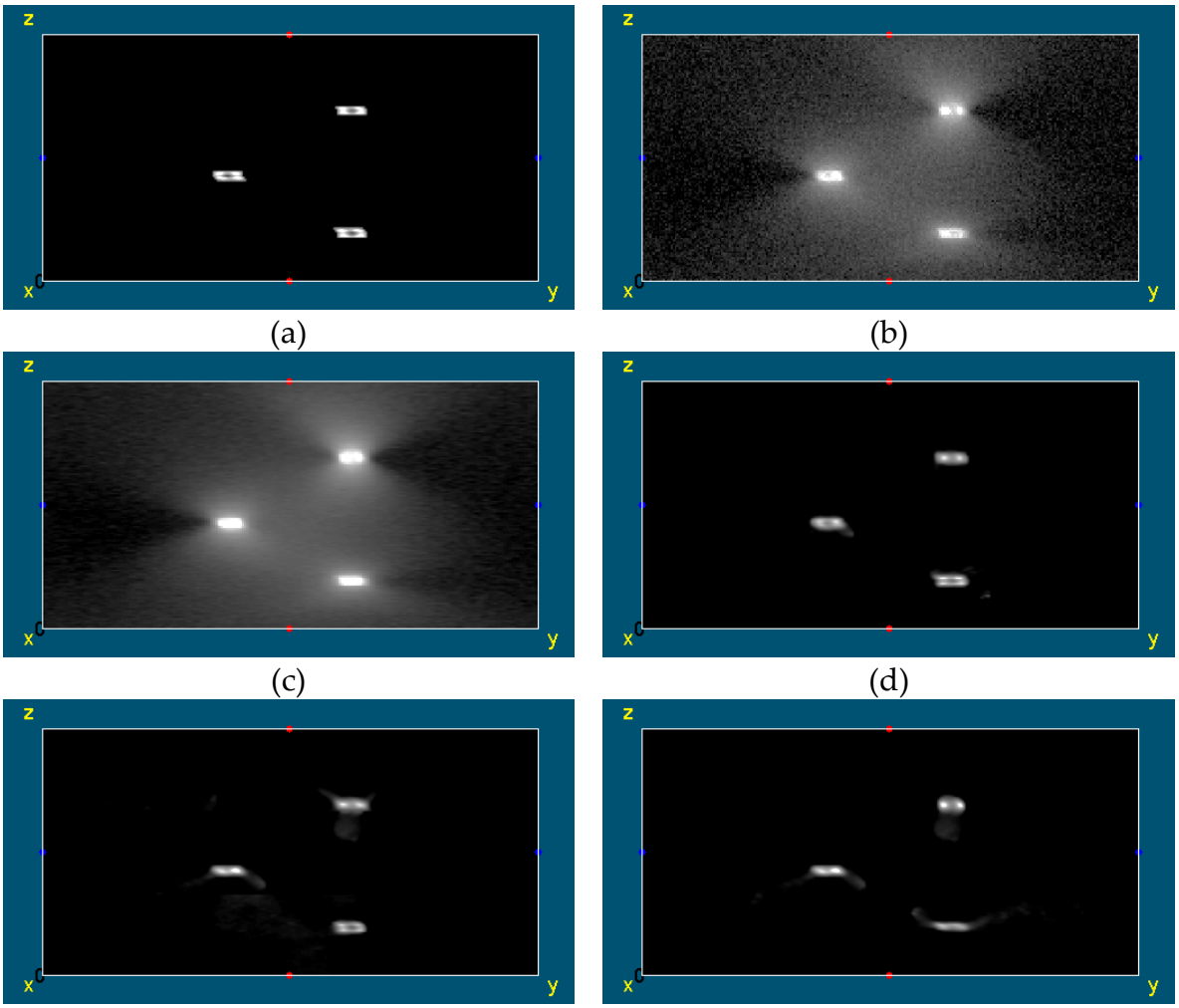


Figure 6.12: Shows an example restoration in the presence of noise. (a) Helix object's *YZ* cross section through its center, (b) 10 dB SNR blurred image, (c) blurred image after smoothing, (d) result of GCT deconvolution, (e) result of PCSI deconvolution and (f) result of SI deconvolution. Refer Table 6.4 for ReMSE.

6.7. 3D shape recovery examples

The focused image (3D) of a surface is a volume object where a point light source of position (x, y) comes to focus at its Gaussian image location along the z axis. Therefore for every (x, y) coordinate position there is only one light source and zero everywhere else. Two such surface objects are considered: spherical surface and an inclined plane. The surface objects were blurred by the Gaussian PSF. Recovering shape from 3D blurred image amounts to restoring impulses along the z direction. After recovering the focused image estimate the location of the maximum intensity for each (x, y) is collected. This is the shape describing function of the surface object.

First example is an inclined plane as shown in Figs. 6.13 and 6.14. Fig. 6.13 shows mesh-grid of the surface. The plane was texture mapped with random dot pattern. Imaging parameters used were $f_0 = 10\,mm,\ F/\# = 5.6$ and $[a, b] = [15, 50]$. The ReMSE between the original and recovered shape functions was 0.0044.

Figs. 6.15 and 6.16 show shape recovery of a spherical surface. The imaging parameters used were $f_0 = 10\,mm, F/\# = 4.0$ and $[a, b] = [25, 50]$. The ReMSE between the original and recovered shape functions was 0.0158.

The GCT based method was not able to recover the texture information satisfactorily (ReMSE ~ 0.75). The shape recovery results are encouraging enough to motivate further investigations on GCT based shape recovery techniques.

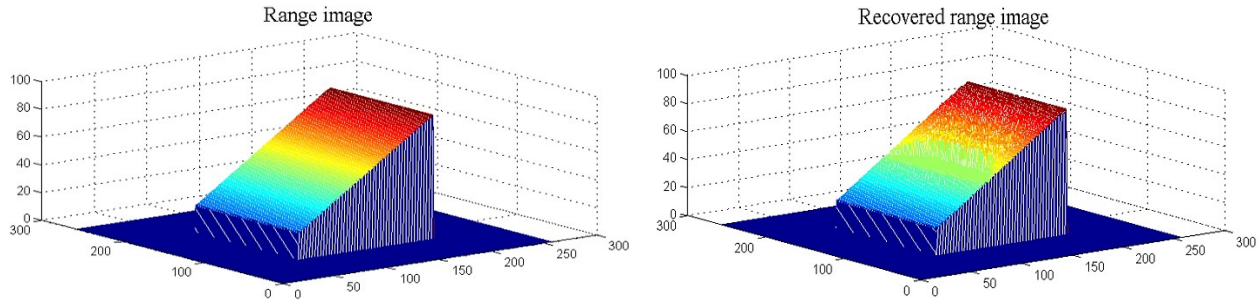


Figure 6.13: Shows the mesh grid plots of original and recovered surfaces.

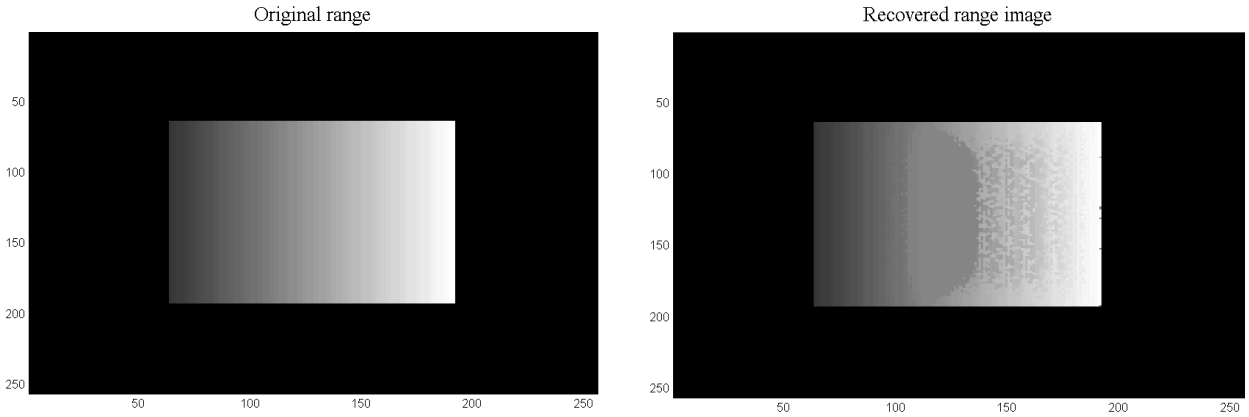


Figure 6.14: Range images of original and recovered planar object.

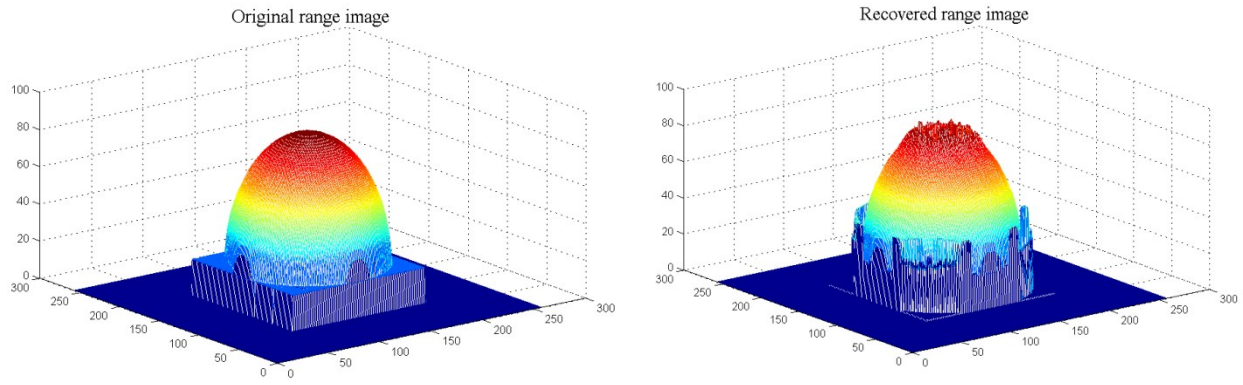


Figure 6.15: Shows the mesh grid plots of spherical object example

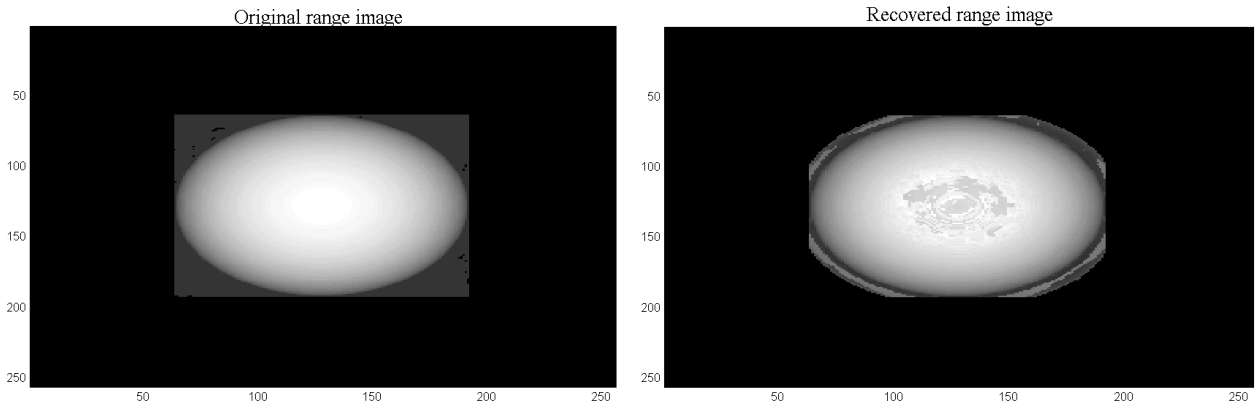


Figure 6.16: Range images of original and recovered planar object.

6.8. Conclusions

This chapter presented application of GCT in 3D restoration of 3D blurred images. Two applications were considered: 3D microscopy and shape recovery. GCT was used to restore the 3D blurred images under conditions that made the 3D imaging system shift-variant along the axial direction. Using GCT, such shift-variant systems were transformed to shift-invariant systems that improved both the accuracy and computation time of restoration. In 3D microscopy application, GCT deconvolution was compared with SI and PCSI deconvolution methods. It was demonstrated that GCT recovers more accurate focused images compared to SI and PCSI methods. GCT was also about a factor of k faster than k-block PCSI deconvolution method. Noisy blurred images were also considered for restoration. A filtered Lucy-Richardson algorithm was employed to restore noisy blurred images in which the noisy images were smoothed before performing restoration. GCT based deconvolution provided accurate and high quality estimates and avoided blocking artifacts. Although shape recovery was possible,

more investigations are needed in the direction of texture recovery from blurred images through GCT.

Possible future applications of GCT in three-dimensional imaging are in restoration of confocal microscope images and restoration of depth dependent spherical aberration in Widefield Microscopes. Some details regarding this are provided in the next chapter. It is also of interest to investigate application of GCT to other fields.

7. Conclusions and Future Work

In this work, computationally efficient algorithms to restore shift-variant blurred images have been discussed. The algorithms in 2D exploited the local nature of point spread functions to derive iterative update equations that can efficiently restore blurred images. Three algorithms MRT, RTI and RTLW methods were discussed in detail. The methods were compared to another localized approach the SRT method for accuracy, quality and efficiency. The MRT method was demonstrated to be effective in restoring highly defocused images. However, its computational cost was higher than the SRT method. The other two methods are iterative approaches using localization as opposed to direct inversion. Through experiments it was demonstrated that the RTI and RTLW methods provide rapid convergence, with usually solutions (best MSE) obtained after three to four iterations.

Of all the methods considered in this work, the MRT method most effective in restoring sharp edges in highly defocused images. The RTI method was the fastest method in all experiments. The comparisons with the exact Landweber's iteration and Tikhonov regularization techniques showed gains in computational efficiency of RTI and RTLW methods. Among these algorithms RTI and RTLW provided the best tradeoff between accuracy, computational efficiency and image quality. Experiments

with noisy input in Chapter 4 show that for low-medium levels of noise, RTI and RTLW methods perform satisfactorily. The new insight due to the localized update equation provided the perspective to handle noisy input. Frequency domain filtering for differentiation was suggested as a way to control noise amplification.

A new theorem called the Generalized Convolution Theorem (GCT) was introduced in Chapter 5. GCT provides the conditions under which a superposition integral can be transformed to a convolution integral. Proofs for one dimensional and multi-dimensional GCT were provided. GCT is applied to linear shift-variant systems to represent them as linear shit-invariant systems in the new space. The resulting convolution equation is computationally efficient to implement and thus shift-variant restoration is greatly simplified by GCT.

The shift-variant kernels of a 3D imaging systems under geometric optics satisfy the required conditions. GCT was applied to 3D microscopy to the shift-variant blur along the *z* axis. The new shift-invariant system was solved to obtain solution to the focused image in the new space. It was then transformed to the original space. Using GCT, it was demonstrated that the computational cost of shift-variant restoration became comparable to shift-invariant restoration (Deconvolution). Results obtained by using GCT were more accurate than shift-invariant and piecewise shift-invariant approximations. GCT was considerably faster than PCSI approach to restoration. Results of initial investigations in application of GCT for shape recovery were also presented. These results provide a glimpse of the possibilities for other applications that solve integral equations.

7.1. Future work

Although the algorithms discussed in this work perform satisfactorily, some improvements are possible. One such improvement is possible for the MRT method for achieving better computational efficiency by observing the RTI method. The *forward* MRT equation can be rearranged to produce an iterative update formula that is a multi-interval version of the RTI algorithm. This approach should be explored as a way to efficiently restore highly defocused images. The other extensions include restoring shift-variant motion blur and degradations due to primary aberrations.

The investigation involving GCT is in its initial stage. In 3D microscopy, there are at least two extensions possible. First is in incorporating wave optics PSFs derived from pupil functions. According to the PSFs developed in Section 6.4.4 of Ref. [19], the aberration function satisfies the conditions required to apply GCT. The other possibility is in deconvolution of confocal microscopy images. A Confocal Microscope's PSF has local support in the z variable as well, unlike the PSFs considered in Chapter 6. It may be possible to incorporate localized approaches to efficiently restore confocal microscope images.

The principle behind other imaging modalities such as Radar imaging is similar to that of optical imaging. It is interesting to investigate applications of GCT to other image modalities.

References

[1] D. Fredo and R. Szeliski. (2007) Guest Editors' Introduction: Computational Photography. 21-22. Available: http://doi.ieeecomputersociety.org/10.1109/MCG.2007.35

[2] M. Levoy, Z. Zhang, and I. McDowall, "Recording and controlling the 4D light field in a microscope using microlens arrays," *Journal of Microscopy,* vol. 235, pp. 144-162, 2009.

[3] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-Held Plenoptic Camera," 2005.

[4] D. A. Agard and J. W. Sedat, "Three-dimensional architecture of a polytene nucleus," *Nature,* vol. 302, pp. 676-681, 1983.

[5] A. F. Boden, D. C. Redding, R. J. Hanisch, and J. Mo, "Massively parallel spatially variant maximum-likelihood restoration of Hubble Space Telescope imagery," *J. Opt. Soc. Am. A,* vol. 13, pp. 1537-1545, 1996.

[6] H. M. Adorf, R. N. Hook, and L. B. Lucy, "HST image restoration developments at the ST-ECF," *International Journal of Imaging Systems and Technology,* vol. 6, pp. 339-349, 1995.

[7] S. Bikkannavar and D. Redding, "The end of the blur," *Spectrum, IEEE,* vol. 47, pp. 46-53, 2010.

[8] M. Subbarao, W. Tse-Chung, and G. Surya, "Focused image recovery from two defocused images recorded with different camera settings," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 1994, pp. 786-791.

[9] A. N. Rajagopalan and S. Chaudhuri, "An MRF Model-Based Approach to Simultaneous Recovery of Depth and Restoration from Defocused Images," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 21, pp. 577-589, 1999.

[10] T. Xue, M. Subbarao, and K. Youn-Sik, "A new approach to 3D shape recovery of local planar surface patches from shift-variant blurred images," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1-5.

[11] X. Tu, Y.-s. Kang, and M. Subbarao, *Depth and focused image recovery from defocused images for cameras operating in macro mode* vol. 6762: SPIE, 2007.

[12] R. L. Lagendijk, A. K. Katsaggelos, and J. Biemond, "Iterative identification and restoration of images," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, 1988, pp. 992-995 vol.2.

[13] H. J. Trussell and S. Fogel, "Identification and restoration of spatially variant motion blurs in sequential images," *Image Processing, IEEE Transactions on,* vol. 1, pp. 123-126, 1992.

[14] F. Sroubek and J. Flusser, "Multichannel blind deconvolution of spatially misaligned images," *Image Processing, IEEE Transactions on,* vol. 14, pp. 874-883, 2005.

[15] J. Markham and J.-A. Conchello, "Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur," *J. Opt. Soc. Am. A,* vol. 16, pp. 2377-2391, 1999.

[16] A. N. Tychonoff and V. Y. Arsenin, *Solution of Ill-posed Problems*. Washington: Winston & Sons, 1977.

[17] M. H. H. W. Engl, and A. Neubauer, *Regularization of Inverse Problems*. Dordrecht, The Netherlands: Kluwer,, 2000.

[18] J. d. Gaskill, *Linear Systems, Fourier Transforms, and Optics*: Wiley-Interscience, 1978.

[19] J. Goodman, *Introduction to Fourier Optics*: McGraw-Hill, 1996.

[20] Max Born and Emil Wolf, *Principles of Optics*, Sixth ed.: Cambridge University Press, 1997.

[21] M. Subbarao and M. Lu, *Computer modeling and simulation of camera defocus* vol. 1822: SPIE, 1993.

[22] V. N. Mahajan, *Optical Imaging and Aberrations, Part II. Wave Diffraction Optics* vol. 2: SPIE Publications, 2001.

[23] S. B. Sastry, "Primary Aberrations: An Investigation from the Image Restoration Perspective," M.S., Electrical and Computer Engineering, Stony Brook University, Stony Brook, 2009.

[24] S. K. Mitra, *Digital Signal Processing*, 3rd ed.: McGraw-Hill, 2005.

[25] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete Time Signal Processing*, 2nd ed.: Prentice-Hall, 1999.

[26] K.C. Andrews and B.R.Hunt, *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[27] J. Kamm and J. G. Nagy, *Kronecker product and SVD approximations for separable spatially variant blurs* vol. 3461: SPIE, 1998.
[28] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008.
[29] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*: Academic Press, Inc., 1982.
[30] A. K. Jain, *Fundamentals of digital image processing*: Prentice-Hall, Inc., 1989.
[31] A. K. Katsaggelos, "Recent trends in image restoration and enhancement techniques," in *Circuits and Systems, 1996., IEEE Asia Pacific Conference on*, 1996, pp. 458-459.
[32] L. Landweber, "An Iteration Formula for Fredholm Integral Equations of the First Kind," *American Journal of Mathematics,* vol. 73, pp. 615-624, 1951.
[33] L. Lei and X. Yuanchang, "Adaptive Landweber method to deblur images," *Signal Processing Letters, IEEE,* vol. 10, pp. 129-132, 2003.
[34] E. S. Angel and A. K. Jain, "Restoration of images degraded by spatially varying pointspread functions by a conjugate gradient method," *Appl. Opt.,* vol. 17, pp. 2186-2190, 1978.
[35] J. G. Nagy and D. P. O'Leary, *Fast iterative image restoration with a spatially varying PSF* vol. 3162: SPIE, 1997.
[36] J. Nagy and K. Palmer, "Steepest Descent, CG, and Iterative Regularization of Ill-Posed Problems," *BIT Numerical Mathematics,* vol. 43, pp. 1003-1017, 2003.
[37] R. Vio, J. Nagy, and W. Wamsteker, "Multiple-image deblurring with spatially-variant point spread functions," *A&A,* vol. 434, pp. 795-800, 2005.
[38] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical Journal,* vol. 79, p. 745, 1974.
[39] W. H. Richardson, "Bayesian-Based Iterative Method of Image Restoration," *J. Opt. Soc. Am.,* vol. 62, pp. 55-59, 1972.
[40] P. Favaro and Stefano Soatto, *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*: Springer, 2006.
[41] H. Trussell and B. Hunt, "Sectioned methods for image restoration," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 26, pp. 157-164, 1978.
[42] H. Trussell and B. Hunt, "Image restoration of space variant blurs by sectioned methods," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '78.*, 1978, pp. 196-198.
[43] T. P. Costello and W. B. Mikhael, "Efficient restoration of space-variant blurs from physical optics by sectioning with modified Wiener filtering," *Digital Signal Processing,* vol. 13, pp. 1-22, 2003.
[44] J. G. Nagy, K. Palmer, and L. Perrone, "Iterative Methods for Image Deblurring: A Matlab Object-Oriented Approach," *Numerical Algorithms,* vol. 36, pp. 73-93, 2004.

[45] J. Kamm and J. G. Nagy, "Kronecker product and SVD approximations in image restoration," *Linear Algebra and its Applications,* vol. 284, pp. 177-192, 1998.
[46] P. C. Hansen. (2006, 08/06/2011). *Image Deblurring*. Available: http://www2.imm.dtu.dk/~pch/HNO/
[47] J. G. Nagy and D. P. O'Leary, "Restoring Images Degraded by Spatially Variant Blur," *SIAM J. Sci. Comput.,* vol. 19, pp. 1063-1082, 1998.
[48] M. E. Kilmer and J. G. Nagy, "Kronecker product approximations for dense block Toeplitz-plus-Hankel matrices," *Numerical Linear Algebra with Applications,* vol. 14, pp. 581-602, 2007.
[49] A. A. Sawchuk, "Space-variant image restoration by coordinate transformations," *J. Opt. Soc. Am.,* vol. 64, pp. 138-144, 1974.
[50] G. M. Robbins and T. S. Huang, "Inverse filtering for linear shift-variant imaging systems," *Proceedings of the IEEE,* vol. 60, pp. 862-872, 1972.
[51] B. Saleh and N. Subotic, "Time-variant filtering of signals in the mixed time frequency domain," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 33, pp. 1479-1485, 1985.
[52] N. Bouaynaya and D. Schonfeld, "Theoretical Foundations of Spatially-Variant Mathematical Morphology Part II: Gray-Level Images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, pp. 837-850, 2008.
[53] M. Subbarao, K. Youn-sik, S. Dutta, and T. Xue, "Localized and computationally efficient approach to shift-variant image deblurring," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, 2008, pp. 657-660.
[54] M. Subbarao, "Methods and apparatus for computing the input and output signals of a linear shift-variant system," United States Patent, 2009.
[55] M. Subbarao, "Direct Vision Sensor for 3D Computer Vision, Digital Imaging, and Digital Video," United States Patent, 2009.
[56] S. B. Sastry and M. Subbarao, *Computationally efficient deblurring of shift-variant highly defocused images* vol. 7798: SPIE, 2010.
[57] K. Youn-Sik, "Investigation of a Localized Approach to Shift-Variant Image Restoration and Robust Autofocusing," Ph. D, Electrical and Computer Engineering, Stony Brook University, Stony Brook, 2011.
[58] M. K. Ozkan, A. M. Tekalp, and M. I. Sezan, "POCS-based restoration of space-varying blurred images," *Image Processing, IEEE Transactions on,* vol. 3, pp. 450-454, 1994.
[59] J. B. Sibarita, "Deconvolution microscopy," *Advances in Biochemical Engineering/Biotechnology* vol. 95, pp. 201-243, 2005.
[60] N. Dey, L. Blanc-Féraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, "3D Microscopy Deconvolution using Richardson-Lucy Algorithm with Total Variation Regularization," 2004-00-00 2004.

[61] M. T. Heath, *Scientific Computing: An Introductory Survey*, 2 ed.: McGraw-Hill, 2002.
[62] G. Strang, *Linear algebra and its applications*, 4th ed. Belmont, CA: Thomson, Brooks/Cole, 2006.
[63] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Image Deblurring: Matrices, Spectra and Filtering*: SIAM, 2006.
[64] P. Meer and I. Weiss, "Smoothed differentiation filters for images," *J. Vis. Comun. Image Represent.,* vol. 3, pp. 58-72, 1992.
[65] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Analytical Chemistry,* vol. 36, pp. 1627-1639, 1964.
[66] P. Holborodko. (2009, 08/06/2011). *Noise Robust Gradient Operators*. Available: http://www.holoborodko.com/pavel/image-processing/edge-detection/
[67] R. Jain, R. Kasturi, and B. G. Schunk, *Machine Vision*: McGraw-Hill, 1995.
[68] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision* vol. 2: Prentice Hall, 2002.
[69] R. P. Kanwal, *Linear Integral Equations: Theory and Techniques*, 2nd ed. Boston: Birkhäuser, 1996.
[70] M. Masujima, *Applied Mathematical Methods in Theoretical Physics*: Wiley-VCH, 2005.
[71] B. R. Frieden, "Optical Transfer of the Three-Dimensional Object," *J. Opt. Soc. Am.,* vol. 57, pp. 56-65, 1967.
[72] S. Frisken Gibson and F. Lanni, "Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy," *J. Opt. Soc. Am. A,* vol. 9, pp. 154-166, 1992.
[73] M. Gu, "Three-dimensional space-invariant point-spread function for a single lens," *J. Opt. Soc. Am. A,* vol. 12, pp. 1602-1604, 1995.
[74] J. D. N. Sitter and W. T. Rhodes, "Three-dimensional imaging: a space invariant model for space variant systems," *Appl. Opt.,* vol. 29, pp. 3789-3794, 1990.
[75] G. M. P. Van Kempen, L. J. Van Vliet, P. J. Verveer, and H. T. M. Van Der Voort, "A quantitative comparison of image restoration methods for confocal microscopy," *Journal of Microscopy,* vol. 185, pp. 354-365, 1997.
[76] K. R. Castleman, *Digital Image Processing*: Prentice Hall Professional Technical Reference, 1979.
[77] D. A. Agard, "Optical Sectioning Microscopy: Cellular Architecture in Three Dimensions," *Annual Review of Biophysics and Bioengineering,* vol. 13, pp. 191-219, 1984/06/01 1984.
[78] D. A. Agard, Y. Hiraoka, P. Shaw, and J. W. Sedat, "Fluorescence microscopy in three dimensions," *Methods in cell biology,* vol. 30, pp. 353-77, 1989.

[79] A. Erhardt, G. Zinser, D. Komitowski, and J. Bille, "Reconstructing 3-D light-microscopic images by digital image processing," *Appl. Opt.,* vol. 24, pp. 194-200, 1985.

[80] J. G. McNally, T. Karpova, J. Cooper, and J. A. Conchello, "Three-Dimensional Imaging by Deconvolution Microscopy," *Methods,* vol. 19, pp. 373-385, 1999.

[81] W. Wallace, L. H. Schaefer, and J. R. Swedlow, "A workingperson's guide to deconvolution in light microscopy," *BioTechniques,* vol. 31, pp. 1076-8, 1080, 1082 passim, Nov 2001.

[82] P. Sarder and A. Nehorai, "Deconvolution methods for 3-D fluorescence microscopy images," *Signal Processing Magazine, IEEE,* vol. 23, pp. 32-45, 2006.

[83] C. Vonesch and M. Unser, "A Fast Thresholded Landweber Algorithm for Wavelet-Regularized Multidimensional Deconvolution," *Image Processing, IEEE Transactions on,* vol. 17, pp. 539-549, 2008.

[84] J.-A. Conchello and J. G. McNally, *Fast regularization technique for expectation maximization algorithm for optical sectioning microscopy* vol. 2655: SPIE, 1996.

[85] C. Preza, M. I. Miller, J. L. J. Thomas, and J. G. McNally, "Regularized linear method for reconstruction of three-dimensional microscopic objects from optical sections," *J. Opt. Soc. Am. A,* vol. 9, pp. 219-228, 1992.

[86] P. J. Verveer, M. J. Gemkow, and T. M. Jovin, "A comparison of image restoration approaches applied to three-dimensional confocal and wide-field fluorescence microscopy," *Journal of Microscopy,* vol. 193, pp. 50-61, 1999.

[87] Z. Kam, B. Hanser, M. G. L. Gustafsson, D. A. Agard, and J. W. Sedat, "Computational adaptive optics for live three-dimensional biological imaging," *Proceedings of the National Academy of Sciences,* vol. 98, pp. 3790-3795, March 27, 2001 2001.

[88] A. J. Peter, Ed., *Deconvolution of images and spectra (2nd ed.)*. Academic Press, Inc., 1996, p.^pp. Pages.

[89] B. M. Hanser, M. G. L. Gustafsson, D. A. Agard, and J. W. Sedat, "Phase retrieval for high-numerical-aperture optical systems," *Opt. Lett.,* vol. 28, pp. 801-803, 2003.

[90] C. Preza and J.-A. Conchello, *Image estimation accounting for point-spread function depth cariation in three-dimensional fluorescence microscopy* vol. 4964: SPIE, 2003.

[91] C. Preza and J.-A. Conchello, "Depth-variant maximum-likelihood restoration for three-dimensional fluorescence microscopy," *J. Opt. Soc. Am. A,* vol. 21, pp. 1593-1601, 2004.

[92] J. G. McNally, C. Preza, J.-A. Conchello, and L. J. Thomas, "Artifacts in computational optical-sectioning microscopy," *J. Opt. Soc. Am. A,* vol. 11, pp. 1056-1067, 1994.