UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND
IMAGE COMPUTING

Master's Thesis

Efficient Computational Approaches for Multiframe Blind Deconvolution

Helen Schomburg

Advisors

Prof. Dr. Bernd Fischer
Institute of Mathematics and Image Computing
University of Lübeck

Prof. Dr. James G. Nagy
Department of Mathematics and Computer Science
Emory University

März 2012

MIC

IM FOCUS DAS LEBEN

I hereby affirm, that I have independently composed this thesis, and that I have not used any aid or sources other than those indicated. This work has not been presented to any other examination board.

Lübeck, March 8, 2012

Abstract

The reconstruction of objects from blurry images has a wide range of applications, for instance in astronomy and biomedical imaging. Assuming that the blur is spatially invariant, image blur can be defined as a two-dimensional convolution between true image and a point spread function. Hence, the corresponding deblurring operation is formulated as an inverse problem called *deconvolution*. Often, not only the true image is unknown, but also the available information about the point spread function is insufficient resulting in an extremely underdetermined *blind deconvolution* problem. Considering multiple blurred images of the object to be reconstructed, leading to a *multiframe blind deconvolution* problem, reduces underdeterminedness. To further decrease the number of unknowns, we transfer the multiframe blind deconvolution problem to a compact version based upon [18] where only one point spread function has to be identified. Applying Givens rotations to this so called *compact multiframe blind deconvolution* problem we derive a new expression, the *compact singleframe blind deconvolution* problem, to significantly speed up computations. We examine both a separate approach applying the APEX method [6] to solve for the point spread function and subsequently reconstruct the true image, and iterative methods, in particular the Gauss-Newton method, to simultaneously solve the underlying problem. For regularization, the Tikhonov method with generalized cross-validation is applied. We implement deconvolution in MATLAB and present results using artificial data where the blur is caused by Gaussian functions.

Contents

1. Introduction

1.1. The Importance of Deblurring Images

Image blur is a widely spread problem present in image data observed by devices such as ground-based telescopes, microscopes, and medical imaging methods, complicating further processing and interpretation. Image deblurring, a postprocessing technique, seeks to reconstruct the sharp image to improve accuracy of blurry image data records. Considering the blur as spatially invariant the degraded image can be described as a two-dimensional convolution of the true image and a point spread function.

Given a blurred image and the point spread function, the true image is detected by solving an inverse problem called *deconvolution* [33, p.59ff.]. The term *blind deconvolution* is used if only the blurred image is given, and the true image is reconstructed without complete knowledge of the point spread function. Blind deconvolution problems are extremely underdetermined, which means that the number of possible solutions is infinite [9, p.529ff.; 22]. Optimization methods applied to solve a blind deconvolution problem are often trapped in local minima leading to insufficient deblurring results. Instead of a single image, utilizing multiple images of the same object blurred with different point spread functions helps to decrease non-uniqueness of the solution. This approach is known as *multiframe blind deconvolution* [8, p.45ff.; 16].

We focus on three major areas of application concerning image deblurring. Although astronomy has predominantly contributed to a growing interest in this field of research over the last decades, deconvolution is also an important tool in biomedical and medical applications such as fluorescence microscopy and computed tomography.

Astronomy Astronomical observations have been part of our entire cultural history making astronomy the oldest field of science. Today, telescopes exploring space are coupled with imaging devices using digital detectors, often CCD image sensors, to record celestial objects. We distinguish between ground-based telescopes located at the Earth's surface, and space telescopes placed outside the Earth's atmosphere. Figure 1.1 gives an overview of how such telescopes operate.

Light waves emitted by a star or another object in space are distorted at the Earth's atmosphere due to atmospheric turbulence. Hence, and because of light pollution on Earth caused by big cities, objects recorded by ground-based telescopes appear

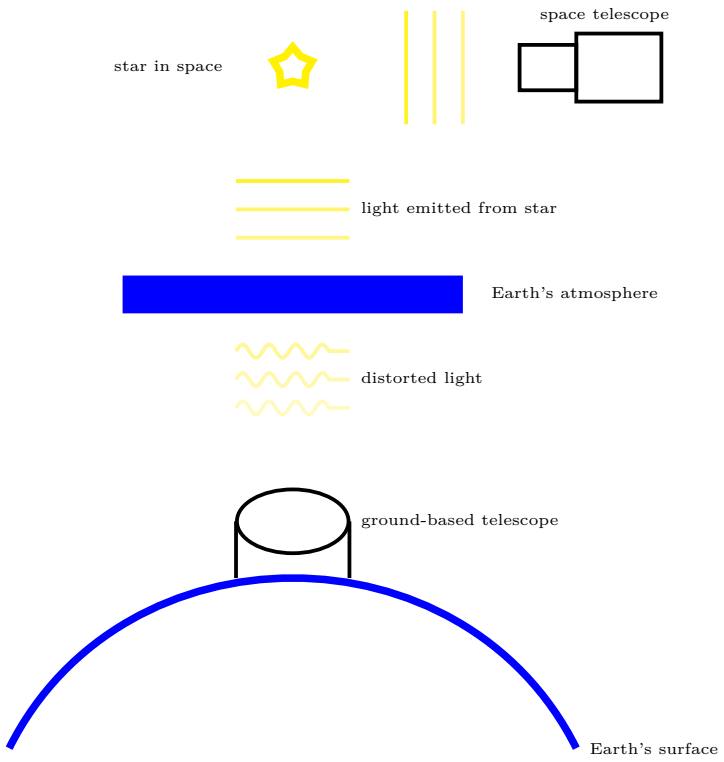


Figure 1.1.: Ground-based telescope and space telescope. The light emitted by objects in space is distorted while traveling through the Earth's atmosphere causing blurred images.

blurred. Blurring effects have been attenuated by manufacturing adaptive optics, and by installing telescopes at high altitude, for instance on Mauna Kea in Hawaii and on high plateaus of the Atacama Desert in Chile. However, computational postprocessing of recorded images is still necessary and rather inexpensive compared to using sophisticated hardware like adaptive optics.

Space Telescopes are located outside Earth's atmosphere, thus able to detect undistorted light waves resulting in less disturbed photographs. Yet postprocessing still remains important, for example if image quality is degraded by technical breakdown or damage. The most famous space telescope is probably the Hubble Space Telescope operating since 1990. In the first few years, it recorded blurry images due to a construction fault. Only preprocessing yielded useful image data records, until repair was possible [1; 3].

An astronomical imaging method using multiframe blind deconvolution is called *speckle imaging*. Here, the object of interest is repeatedly recorded with short exposure time. Because the atmospheric turbulence is constantly changing, each image in this sequence is blurred differently. Multiframe blind deconvolution helps to reconstruct the true image using these different observations, as for instance examined by Schulz [30].

Fluorescence Microscopy The fluorescence microscope is an important tool in biology and biomedicine to visualize subcellular components and their interactions. It is even possible to observe protein-protein interactions. Moreover, three-dimensional views of objects can be generated by examining and merging different image slices of a sample.

In general, fluorescence microscopy operates as follows: Fluorescent dye molecules are added to the object of interest, for instance a medical sample of a patient's tissue. Then, a light source of high energy (e.g. blue light) is directed at the regarded specimen, exciting its electrons to a higher energy state. Fluorescence occurs when the electrons return to their ground state and light of lower energy (e.g. green light) is emitted that can be detected through the microscope's objective [28]. The procedure is illustrated in Figures 1.2 (a) and (b). Since all structures of the specimen emit light, including those that are not in focus, the observed object appears blurred.

Efforts made to technically improve this method include the invention of *confocal microscopy*. This further development of fluorescence microscopy seeks to dilute image blur by sending light through a pinhole in front of the detector, and scanning the sample pointwise instead of illuminating the whole sample at once. The idea of attaching a screen with a pinhole to the microscope is demonstrated in Figure 1.2 (c). In theory, light emitted by unfocused points of the sample cannot pass through the screen and reach the detector, hence only focused points can be observed [20; 29].

However, confocal microscopy does not yield perfect results, as the pinhole would need to be extremely small. Image blur does not entirely vanish and signal intensity is reduced due to installation of the pinhole [31]. Since accuracy is of particular importance in microscopy, especially in medical applications, images are further improved by applying deconvolution techniques. For instance, Cannell et al. [5] explain a way to estimate point spread functions experimentally, whereas Holmes et al. [17], Pankajakshan [26] and Pankajakshan et al. [27] concentrate on blind deconvolution.

Computed Tomography Widely-used for clinical diagnostics and in medical science, computed tomography (CT) is one of the most important medical imaging methods in today's medicine. Figure 1.3 describes the general procedure of image data collection using a CT scanner. X-rays penetrate the patient's body and are detected on the opposite side. Since attenuation of x-rays in the body differs depending on the type of tissue, acquired image data shows a contrast between distinct tissues. For instance, bones, which are less penetrable for radiation, appear white in a CT image. Soft tissue, on the other hand, appears darker, because more radiation passes through. CT images of various thin slices of the patient's body are collected from different angles to reconstruct a three-dimensional volume [35].

The higher the radiation dose an object is exposed to, the higher is the image resolution that can be attained. A low radiation dose results in unsharp, blurry images, whereas a

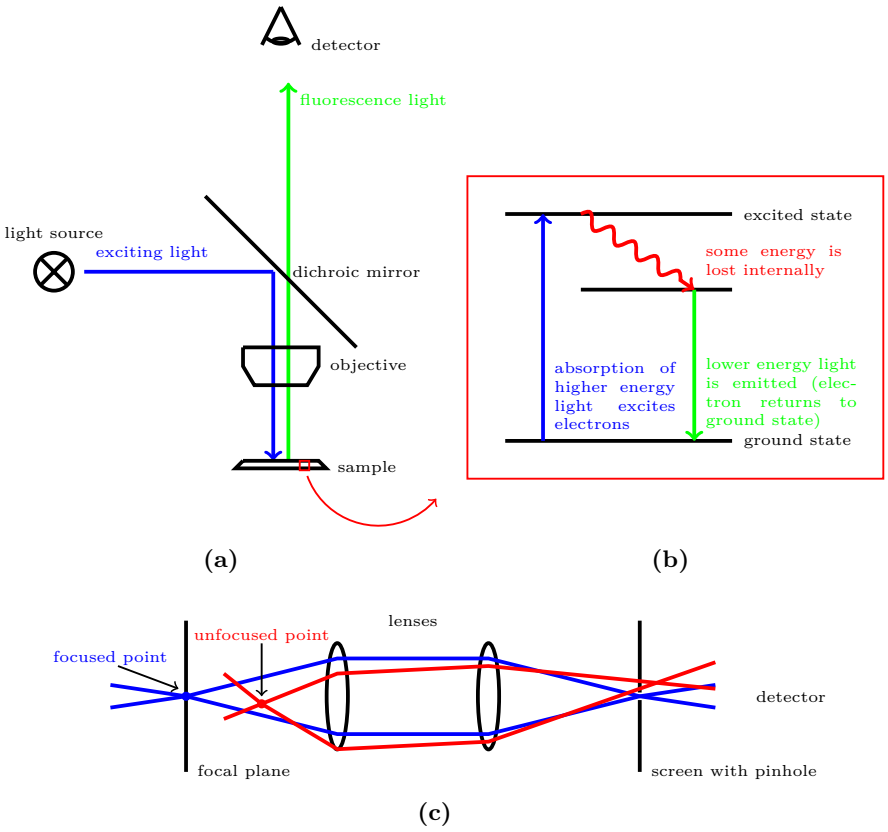


Figure 1.2.: Conceptual diagram of (a) a fluorescence microscope, (b) occurrence of fluorescence due to electrons in the sample changing their energy state, and (c) eliminating the detection of unfocused points by attaching a screen with a pinhole. In (a), a *dichroic mirror* is a device that helps to direct light of different energy. High energy light (e.g. blue light) is reflected, whereas lower energy light (e.g. green light) passes the mirror.

high radiation dose is accompanied by an increasing cancer risk. Therefore, combining low-dose x-rays and image deblurring techniques simultaneously protects the patient from radiation and provides adequately sharp image data.

Apart from that, image deblurring is, regardless of radiation dose, applied to CT images simply for image enhancement. For instance, Jiang et al. [21] successfully use blind image deconvolution to enhance resolution of CT images of the temporal bone to support the installation of cochlear implants. A cochlear implant restores or improves auditory perception in hearing-impaired patients. Prior to surgical implantation into the temporal bone, spiral CT images of this region are produced to help positioning the implant. Here, it is important to distinguish the fine bony structures of middle and inner ear. CT image resolution, however, is too low to show some important features. Therefore, applying image deconvolution to reveal details is crucial to safely implant the device.

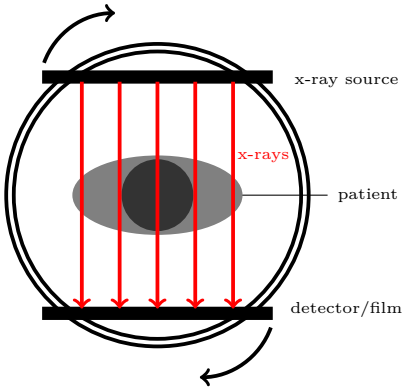


Figure 1.3.: The principle of computed tomography.

1.2. How This Work Addresses the Problem

This master's thesis was developed as part of a one-year study visit at Emory University in Atlanta/USA. Supervised by Professor James G. Nagy and Professor Bernd Fischer, it is a joint work between the Department of Mathematics and Computer Science at Emory University, and the Institute of Mathematics and Image Computing at the University of Lübeck.

In many applications, an experimental derivation of the point spread function is imprecise or impossible resulting in a blind deconvolution problem. Underdeterminedness of this problem can be reduced by setting up a multiframe blind deconvolution problem. In addition, real image data, for instance acquired by astronomical, microscopical, or medical imaging devices as discussed above, usually consists of large images. Both, increasing problem size due to multiple images, and large image size, involve relatively long processing times and large memory requirements.

The aim of this work is to identify strategies for efficiently performing multiframe blind deconvolution. Our approaches are based on the papers by Hope and Jefferies [18] and Carasso [6]. We transform the compact multiframe blind deconvolution problem stated by Hope and Jefferies into a singleframe problem to significantly speed up computations. Moreover, combining the derived singleframe problem with one of Carasso's methods, we present a direct solution method.

The thesis is structured as follows:

Chapter 2 deals with the basic principles of image deblurring. We consider the blurring process as a two-dimensional convolution of true image and point spread function, and describe its conversion to a matrix-vector multiplication. Subsequently, a strategy to reconstruct the true image, given the blurred image and the point spread function, is presented. Moreover, we give an overview of multiframe deconvolution and blind deconvolution, two topics that are regarded in more detail in the following chapters.

Chapter 3 focuses on solving the extremely underdetermined blind deconvolution problem, where only information about the blurred image is available. Here, two iterative strategies as well as a direct method to obtain the deblurred image are examined.

Chapter 4 introduces the multiframe blind deconvolution problem. It is described how the expression of this problem can be simplified under certain assumptions yielding a more compact version. Moreover, we show that the problem can be further reduced to a singleframe blind deconvolution problem applying Givens rotations.

Chapter 5 presents results from multiframe blind deconvolution of simulated data. Combining information from Chapters 3 and 4, we compare iteratively obtained results of the three regarded versions of the multiframe problem, and directly reconstruct the sharp image from the singleframe blind deconvolution problem derived in Chapter 4.

Chapter 6 summarizes important aspects, discusses which conclusions can be drawn from this work, and identifies potential future research.

2. Fundamentals of Image Deblurring

Unintentional image blur can have various sources. Both the recording device itself, for example an unfocused camera, and external interferences, like atmospheric turbulence or movement, result in an inaccurate, blurred image instead of depicting the real sharp object. In this chapter we give an introduction to image deblurring, based upon the book by Hansen et al. [16]. More precisely, we state a mathematical model for the blurring process that involves so called *point spread functions* and review possible ways of solving the deblurring problem, which is the reconstruction of the sharp image using knowledge about blurred image and the blurring process. Moreover, we consider fundamentals of more advanced topics like *multiframe* image deblurring, where several different blurred images of the same object are used to restore the desired sharp image, and *blind* image deblurring, where only the blurred image but not the blurring process is known.

2.1. A Mathematical Model

We define the blurring process, illustrated in Figure 2.1, as a two-dimensional convolution

$$\begin{aligned} B(s,t) &= P(s,t) * X(s,t) \\ &= \int_{\mathbb{R}^2} P(s',t')X(s-s',t-t')ds'dt', \end{aligned} \tag{2.1}$$

where the functions B, P and $X : \mathbb{R}^2 \to \mathbb{R}_+$ represent the blurred image, point spread function and true image, respectively. The point spread function P that smears the contours in the true image X cannot be stated as one specific function, since it depends on the particular source of the blurring. In the next section we go into more detail in terms of point spread functions.

In the following we consider images as arrays where each entry represents a pixel of a certain gray value. This interpretation requires a discretized expression of (2.1), given by

$$\mathbf{B} = \mathbf{P} * \mathbf{X}$$

where $\mathbf{B}$, $\mathbf{P}$ and $\mathbf{X}$ are arrays of the same size. Each array element $b_{ij} \in \mathbb{R}_+$ of $\mathbf{B}$,

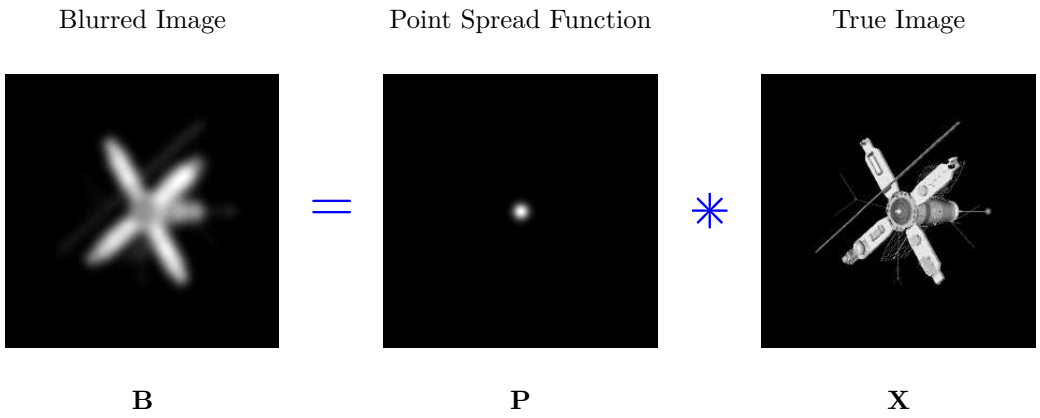


Figure 2.1.: The blurring process.[1]

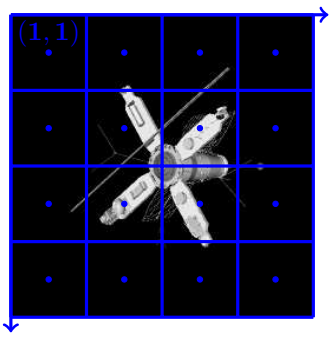


Figure 2.2.: Simplified description of an image array composed of single pixels. Each square represents one pixel, where the dot in the center denotes the pixel value, which is an average of the intensities in the square. From the left upper corner where $i = 1$ and $j = 1$, the indices increase from left to right and from top to bottom, respectively.

identified by subscripts i and j, is computed as

$$b_{ij} = \sum_{i'=-\infty}^{\infty} \sum_{j'=-\infty}^{\infty} p_{i'j'} x_{(i-i',j-j')}. \tag{2.2}$$

where $p_{ij} \in \mathbb{R}_+$ and $x_{ij} \in \mathbb{R}_+$ are elements of $\mathbf{P}$ and $\mathbf{X}$, respectively. However, we only want to regard image arrays containing a finite number of pixels where $i = 1, \ldots, m$ and $j = 1, \ldots, n$ as indicated in Figure 2.2.

We illustrate the two-dimensional discrete convolution of two arrays with a finite num-

[1] The satellite image, that appears throughout this thesis, originates from the US Air Force Phillips Laboratory, Lasers and Imaging Directorate, Kirtland Air Force Base, New Mexico. It has been widely used in literature for testing image deconvolution algorithms.

ber of elements by means of an example. Let

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}, \mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix}, \text{ and } \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

denote the discrete 3×3 true image, point spread function array and blurred image, respectively. Moreover, we define

$$\mathbf{P}_{\curvearrowleft} = \begin{pmatrix} p_{33} & p_{32} & p_{31} \\ p_{23} & p_{22} & p_{21} \\ p_{13} & p_{12} & p_{11} \end{pmatrix}$$

as the array $\mathbf{P}$, rotated by $180°$. Then, to compute entry b_{ij} by convolving $\mathbf{X}$ and $\mathbf{P}$, we position array $\mathbf{P}_{\curvearrowleft}$ on array $\mathbf{X}$ so that the center of $\mathbf{P}_{\curvearrowleft}$, p_{22}, is located on top of x_{ij} and add up the products of directly superposed elements. Since on the borders of $\mathbf{X}$, not every element in $\mathbf{P}_{\curvearrowleft}$ covers an element in $\mathbf{X}$, we have to impose certain *boundary conditions* to avoid artifacts near the borders.

One possibility is to suppose that all elements besides x_{ij} are zero. This is referred to as *zero boundary conditions*, and we expand $\mathbf{X}$ to

$$\mathbf{X}_{\text{zero}} = \left(\begin{array}{c|ccc|c} 0 & 0 & 0 & 0 & 0 \\ \hline 0 & x_{11} & x_{12} & x_{13} & 0 \\ 0 & x_{21} & x_{22} & x_{23} & 0 \\ 0 & x_{31} & x_{32} & x_{33} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right),$$

in the 3×3 example. Then, element b_{11} of $\mathbf{B}$ is computed by

$$\begin{aligned} b_{11} = {} & p_{33} \cdot 0 + p_{32} \cdot 0 + p_{31} \cdot 0 \\ & + p_{23} \cdot 0 + p_{22} x_{11} + p_{21} x_{12} \\ & + p_{13} \cdot 0 + p_{12} x_{21} + p_{11} x_{22}. \end{aligned}$$

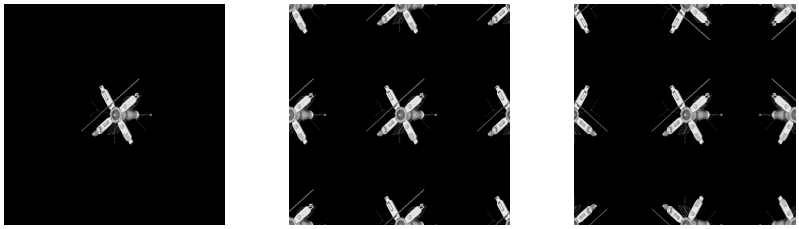
The remaining b_{ij}, $i, j = 1, \ldots, 3$ can be obtained in a similar manner.

Periodic boundary conditions assume that the image $\mathbf{X}$ is circularly repeated beyond its borders, like

$$\mathbf{X}_{\text{periodic}} = \left(\begin{array}{c|ccc|c} x_{33} & x_{31} & x_{32} & x_{33} & x_{31} \\ \hline x_{13} & x_{11} & x_{12} & x_{13} & x_{11} \\ x_{23} & x_{21} & x_{22} & x_{23} & x_{21} \\ x_{33} & x_{31} & x_{32} & x_{33} & x_{31} \\ \hline x_{13} & x_{11} & x_{12} & x_{13} & x_{11} \end{array}\right)$$

for a 3×3 image. Here, the first component of $\mathbf{B}$ is given by

$$\begin{aligned} b_{11} = {} & p_{33} x_{33} + p_{32} x_{31} + p_{31} x_{32} \\ & + p_{23} x_{13} + p_{22} x_{11} + p_{21} x_{12} \\ & + p_{13} x_{23} + p_{12} x_{21} + p_{11} x_{22}. \end{aligned}$$



(a) Zero **(b)** Periodic **(c)** Reflexive

Figure 2.3.: Intended expansion of $\mathbf{X}$ if certain boundary conditions are imposed.

Using *reflexive boundary conditions*, the continuation beyond the boundaries of $\mathbf{X}$ is the mirror image. Again, we give an illustration with respect to the 3×3 case, which can be written as

$$\mathbf{X}_{\text{reflexive}} = \left(\begin{array}{c|ccc|c} x_{11} & x_{11} & x_{12} & x_{13} & x_{13} \\ \hline x_{11} & x_{11} & x_{12} & x_{13} & x_{13} \\ x_{21} & x_{21} & x_{22} & x_{23} & x_{23} \\ x_{31} & x_{31} & x_{32} & x_{33} & x_{33} \\ \hline x_{31} & x_{31} & x_{32} & x_{33} & x_{33} \end{array}\right).$$

Similar to the previous examples,

$$\begin{aligned} b_{11} = p_{33}x_{11} + p_{32}x_{11} + p_{31}x_{12} \\ + p_{23}x_{11} + p_{22}x_{11} + p_{21}x_{12} \\ + p_{13}x_{21} + p_{12}x_{21} + p_{11}x_{22} \end{aligned}$$

illustrates how the first element of the blurred image is computed.

The 3×3 case stated above can be straightforward transferred to larger matrices. Here, an image array of size $m \times n$ is expanded by $\lfloor \frac{m}{2} \rfloor$ in one, and by $\lceil \frac{m}{2} \rceil - 1$ in the other vertical direction. Horizontally, $\lfloor \frac{n}{2} \rfloor$ columns are added on one side, and $\lceil \frac{n}{2} \rceil - 1$ on the other. We illustrate the three different types of boundaries discussed above in Figure 2.3.

2.2. The Point Spread Function

In this section we take a closer look at point spread functions that were introduced in the previous section as provoking blurred images. We regard a discrete point spread function (PSF) as an image that contains pixels with different gray values between 0 and 1 and whose overall entries sum up to one, that is

$$\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} = 1, \text{ where } p_{ij} \in [0, 1].$$

Blur caused by atmospheric turbulence can be estimated by *Gaussian blur*, a PSF whose explicit expression is known. The discrete, two-dimensional Gaussian PSF that is given by

$$p_{ij} = \frac{1}{2\pi\sqrt{\delta}} \exp\left\{-\frac{1}{2}\begin{pmatrix} i-k \\ j-l \end{pmatrix}^T \mathbf{E}^{-1} \begin{pmatrix} i-k \\ j-l \end{pmatrix}\right\},$$

where

$$\mathbf{E} = \begin{pmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{pmatrix} \text{ and } \delta = |\mathbf{E}| = s_1^2 s_2^2 - \rho^4,$$

varies for different values of s_1 and s_2 in width whereas parameter ρ determines its orientation. Moreover, (k, l), $1 \leqslant k \leqslant m$, $1 \leqslant l \leqslant n$ describes the *center* of the PSF, that is the coordinates of the brightest pixel or, equivalently, the maximum entry. If $\rho = 0$, the formula is

$$p_{ij} = \frac{1}{2\pi s_1 s_2} \exp\left\{-\frac{1}{2}\left(\frac{(i-k)^2}{s_1^2} + \frac{(j-l)^2}{s_2^2}\right)\right\}$$

that is the PSF has a vertical and a horizontal symmetry axis. Furthermore, if $s_1 = s_2 = s$ and $\rho = 0$, then the PSF is rotationally symmetric and we get

$$p_{ij} = \frac{1}{2\pi s^2} \exp\left\{-\frac{1}{2s^2}\left((i-k)^2 + (j-l)^2\right)\right\}.$$

Figure 2.4 illustrates how the choice of these parameters impacts the PSF image and the resulting blurred image.

2.3. Matrix-Vector Representation

In the following, we consider the true image and blurred image as real-valued, two-dimensional arrays of size $m \times n$ whose entries represent certain gray values. We define a vector notation, where the columns of a two-dimensional array are arranged one below the other, as $\text{vec}(\mathbf{X}) = \mathbf{x} \in \mathbb{R}_+^{mn\times 1}$ and $\text{vec}(\mathbf{B}) = \mathbf{b} \in \mathbb{R}_+^{mn\times 1}$. The convolution (2.2) can be replaced by a matrix-vector multiplication

$$\mathbf{b} = \text{vec}(\mathbf{P} * \mathbf{X}) = \mathbf{A}\mathbf{x} \tag{2.3}$$

where matrix $\mathbf{A} \in \mathbb{R}_+^{mn\times mn}$ describes the blur and differs according to the imposed boundary conditions introduced in Section 2.1.

Using this representation we can easily formulate the deblurring problem as a *linear least squares* problem, that is finding vector $\mathbf{x}$ with

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \tag{2.4}$$

given $\mathbf{b}$ and $\mathbf{A}$ (see for example the book by Nocedal and Wright [25, p.250]). After discussing the structure of matrix $\mathbf{A}$ for different boundary conditions we look at approaches to solve the classic deblurring problem.

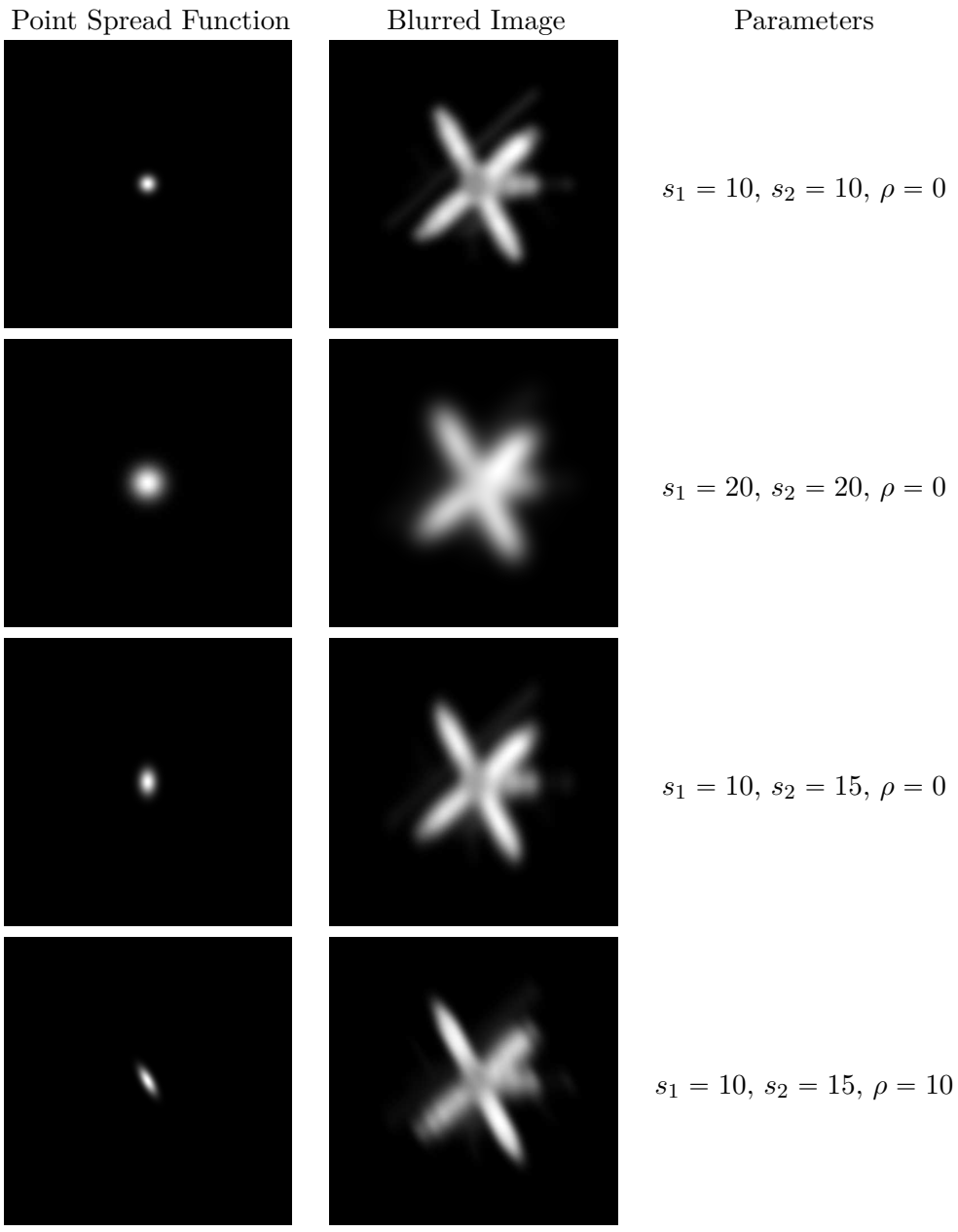


Figure 2.4.: Appearance of Gaussian PSFs for different choice of parameters s_1, s_2 and ρ and the corresponding blurred image (image size: 512×512).

2.3.1. Zero Boundary Conditions

Continuing the example in Section 2.1 illustrating discrete convolution of two arrays of size 3×3, the blurred image in vector form applying zero boundary conditions

$$\underbrace{\begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{31} \\ b_{32} \\ b_{33} \end{pmatrix}}_{\mathbf{b}} = \underbrace{\left(\begin{array}{ccc|ccc|ccc} p_{22} & p_{12} & & p_{21} & p_{11} & & & & \\ p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} & & & \\ & p_{32} & p_{22} & & p_{31} & p_{21} & & & \\ \hline p_{23} & p_{13} & & p_{22} & p_{12} & & p_{21} & p_{11} & \\ p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} \\ & p_{33} & p_{23} & & p_{32} & p_{22} & & p_{31} & p_{21} \\ \hline & & & p_{23} & p_{13} & & p_{22} & p_{12} & \\ & & & p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} \\ & & & & p_{33} & p_{23} & & p_{32} & p_{22} \end{array}\right)}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{pmatrix}}_{\mathbf{x}}$$

forms as matrix-vector product of a block Toeplitz matrix with Toeplitz blocks (BTTB) with vector $\mathbf{x}$. In general, an $N \times N$ *Toeplitz matrix* [19], characterized by constant diagonals, is defined as a matrix of the form

$$\mathbf{T} = \begin{pmatrix} t_0 & t_1 & t_2 & \dots & \dots & t_{N-1} \\ t_{-1} & t_0 & t_1 & \ddots & & t_{N-2} \\ t_{-2} & t_{-1} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & t_1 & t_2 \\ t_{-(N-2)} & & \ddots & t_{-1} & t_0 & t_1 \\ t_{-(N-1)} & \dots & \dots & t_{-2} & t_{-1} & t_0 \end{pmatrix}. \tag{2.5}$$

Various methods have been developed to efficiently solve a least squares problem $\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2$ where $\mathbf{A}$ is a BTTB matrix, and there exist algorithms for easy decomposition of $\mathbf{A}$. See for instance the research done by Chan and Jin [7], Ng [24], and Wax and Kailath [34]. However, zero boundary conditions are only an adequate choice if the object in image $\mathbf{X}$ has a black background, such that the regions close to the boundaries are zero. In the following, we consider two boundary conditions that model the regions outside the image boundaries more realistically.

2.3.2. Periodic Boundary Conditions

Periodic boundary conditions yield a block circulant matrix $\mathbf{A}$ with circulant blocks (BCCB) that has to be substituted in representation (2.3). For instance, a blurred

3×3 image can be obtained by a matrix-vector product

$$\underbrace{\begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{31} \\ b_{32} \\ b_{33} \end{pmatrix}}_{\mathbf{b}} = \underbrace{\left(\begin{array}{ccc|ccc|ccc} p_{22} & p_{12} & p_{32} & p_{21} & p_{11} & p_{31} & p_{23} & p_{13} & p_{33} \\ p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} & p_{33} & p_{23} & p_{13} \\ p_{12} & p_{32} & p_{22} & p_{11} & p_{31} & p_{21} & p_{13} & p_{33} & p_{23} \\ \hline p_{23} & p_{13} & p_{33} & p_{22} & p_{12} & p_{32} & p_{21} & p_{11} & p_{31} \\ p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} \\ p_{13} & p_{33} & p_{23} & p_{12} & p_{32} & p_{22} & p_{11} & p_{31} & p_{21} \\ \hline p_{21} & p_{11} & p_{31} & p_{23} & p_{13} & p_{33} & p_{22} & p_{12} & p_{32} \\ p_{31} & p_{21} & p_{11} & p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} \\ p_{11} & p_{31} & p_{21} & p_{13} & p_{33} & p_{23} & p_{12} & p_{32} & p_{22} \end{array}\right)}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{pmatrix}}_{\mathbf{x}}$$

between the vector representation of true image $\mathbf{X}$ and the BCCB matrix consisting of a structured composition of elements of the PSF array.

Commonly, a circulant matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$ is described, like for example in the book by Horn and Johnson [19], by

$$\mathbf{C} = \begin{pmatrix} c_1 & c_2 & \dots & \dots & c_N \\ c_N & c_1 & c_2 & \dots & c_{N-1} \\ c_{N-1} & c_N & c_1 & \dots & c_{N-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c_2 & c_3 & \dots & c_N & c_1 \end{pmatrix}$$

or, more precisely, row i, $i = 1, \dots, N$ is a cyclic permutation of row $i = 1$ with offset $i - 1$. Thus, the whole matrix is determined by its first row. Every circulant matrix is normal, that is $\mathbf{C}^T\mathbf{C} = \mathbf{C}\mathbf{C}^T$, which implies that its spectral decomposition exists. In particular, the spectral decomposition is given by

$$\mathbf{C} = \mathbf{F}_N^* \mathbf{\Lambda}_\mathbf{C} \mathbf{F}_N,$$

where $\mathbf{\Lambda}_\mathbf{C}$ is a diagonal matrix

$$\mathbf{\Lambda}_\mathbf{C} = \operatorname{diag}(\boldsymbol{\lambda}_\mathbf{C}), \quad \boldsymbol{\lambda}_\mathbf{C} = (\lambda_1, \lambda_2, \dots, \lambda_N)^T,$$

containing the eigenvalues of $\mathbf{C}$, and $\mathbf{F}_N \in \mathbb{C}^{N \times N}$ describes the discrete Fourier transform (DFT) matrix [10, p.72]. The $N \times N$ DFT matrix

$$\mathbf{F}_N = \left(\omega_N^{jk}\right)_{j,k=0}^{N-1} = \begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & \dots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \dots & \omega_N^{2(N-1)} \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \dots & \dots & \omega_N^{(N-1)(N-1)} \end{pmatrix}, \tag{2.6}$$

where $\omega_N = e^{-\frac{2\pi i}{N}}$ specifies an Nth root of unity, is used to formulate the discrete Fourier transform as a matrix multiplication, which is for instance explained by Mertins

[23]. The matrix $\mathbf{F}_N^*$ is defined by the conjugate transpose of the DFT matrix multiplied by $\frac{1}{N}$

$$\mathbf{F}_N^* = \left(\frac{\omega_N^{-jk}}{N}\right)_{j,k=0}^{N-1} = \frac{1}{N}\begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & \omega_N & \omega_N^{-2} & \dots & \omega_N^{-(N-1)} \\ 1 & \omega_N^{-2} & \omega_N^{-4} & \dots & \omega_N^{-2(N-1)} \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_N^{-(N-1)} & \dots & \dots & \omega_N^{-(N-1)(N-1)} \end{pmatrix},$$

that is $\mathbf{F}_N^*\mathbf{F}_N = \mathbf{I}_N$. Multiplied by a vector, $\mathbf{F}_N^*$ performs the inverse discrete Fourier transform.

As illustrated above for an image of size 3×3, the PSF matrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \dots & \mathbf{C}_n \\ \mathbf{C}_n & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}_2 \\ \mathbf{C}_2 & \dots & \mathbf{C}_n & \mathbf{C}_1 \end{pmatrix} \in \mathbb{R}_+^{mn \times mn}$$

contains circulant matrices $\mathbf{C}_i \in \mathbb{R}_+^{m\times m}$, $i = 1, \dots, n$ which are themselves arranged circulantly. As in the case of a circulant matrix, a BCCB matrix is still normal and thus diagonalizable by a unitary matrix. According to Davis [10, p.185], $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = (\mathbf{F}_n \otimes \mathbf{F}_m)^* \mathbf{\Lambda}_\mathbf{A} (\mathbf{F}_n \otimes \mathbf{F}_m) \tag{2.7}$$

where $\otimes$ represents the Kronecker product and $(\mathbf{F}_n \otimes \mathbf{F}_m)$ describes the two-dimensional DFT matrix. For convenience, we define $\mathbf{F} := (\mathbf{F}_n \otimes \mathbf{F}_m)$ and denote the matrix containing the eigenvalues of $\mathbf{A}$ by

$$\mathbf{\Lambda} := \mathbf{\Lambda}_\mathbf{A} = \operatorname{diag}(\boldsymbol{\lambda}), \quad \boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_{mn})^T \tag{2.8}$$

since this decomposition of matrix $\mathbf{A}$ is often used in the subsequent sections. As for the one-dimensional discrete Fourier transform, it holds $\mathbf{F}^*\mathbf{F} = \mathbf{I}$ where $\mathbf{F}^*$ is the inverse two-dimensional DFT matrix.

In the following we show that the eigenvalues of $\mathbf{A}$ can be easily obtained. First,

$$\mathbf{A} = \mathbf{F}^*\mathbf{\Lambda}\mathbf{F} \iff \mathbf{F}\mathbf{A} = \mathbf{\Lambda}\mathbf{F}.$$

Examining the DFT matrix in (2.6) we discover that the first column $\mathbf{f}_1$ of $\mathbf{F} \in \mathbb{C}^{mn\times mn}$, the Kronecker product of two one-dimensional DFT matrices, entirely consists of ones. From that it follows

$$\begin{aligned} & \mathbf{F}\mathbf{a}_1 = \mathbf{\Lambda}\mathbf{f}_1 \\ \iff & \mathbf{F}\mathbf{a}_1 = \mathbf{\Lambda}\mathbf{1}_{mn} = \boldsymbol{\lambda}, \end{aligned}$$

where $\mathbf{1}_{mn}$ is a column vector of ones. Hence, we receive the eigenvalues of $\mathbf{A}$ by the Fourier transform of its first column. By inspecting matrix $\mathbf{A}$ one observes that the first column $\mathbf{a}_1$ written columnwise into an array $\mathbf{P}^{\text{circ}} \in \mathbb{R}_+^{m\times n}$ is equivalent to applying a circular shift to the PSF array $\mathbf{P}$. In particular, the rows in $\mathbf{P}$ are circularly shifted $(k-1)$ up and the columns $(l-1)$ to the left where index $(k,l) \in \mathbb{N}^2$ denotes the center of array $\mathbf{P}$. Formally, making use of MATLAB notation to simplify the expression and due to later application, we define

$$\texttt{reshape}(\mathbf{a}_1, m, n) = \mathbf{P}^{\text{circ}} = \texttt{circshift}(\mathbf{P}, 1 - \texttt{[}k, l\texttt{]})$$

where the circular shift is computed componentwise as

$$p_{ij}^{\text{circ}} = p_{\text{mod}(i-(k+1),m)+1,\text{mod}(j-(l+1),n)+1}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

As a result, $\mathbf{b} = \mathbf{F}^*\mathbf{\Lambda}\mathbf{F}\mathbf{x}$ can be computed very efficiently without explicitly constructing matrix $\mathbf{A}$, and using for instance Fast Fourier Transforms (FFT) to perform DFT matrix multiplications. In the following, we denote the Fourier transform of a matrix using the hat accent. For instance, let arrays $\mathbf{X}, \mathbf{B}$ of size $m \times n$ be the true and blurred image and $\mathbf{x} = \text{vec}(\mathbf{X}), \mathbf{b} = \text{vec}(\mathbf{B}) \in \mathbb{R}_+^{mn\times 1}$ the corresponding vector notation, as stated above. Then

$$\begin{aligned} \hat{\mathbf{x}} &:= \mathbf{F}\mathbf{x} \\ \hat{\mathbf{b}} &:= \mathbf{F}\mathbf{b} \end{aligned} \tag{2.9}$$

and reshaped arrays $\hat{\mathbf{X}}$ and $\hat{\mathbf{B}}$ denote the two-dimensional Fourier transform of true and blurred image, respectively.

Similar to inducing zero boundary conditions, assuming an image to be repeated periodically beyond its borders is rather unrealistic for a taken picture. Nonetheless, a great advantage involving periodic boundary conditions is that we can significantly speed up computations by a transfer to the Fourier domain. Due to this property, we make use of periodic boundary conditions in the course of our work towards efficiently deblurring images.

The Gaussian point spread function frequently appears throughout this thesis as a canonical example for the blur in our model problems. In the following, we want to take a closer look at the eigenvalues of the Gaussian PSF. As stated above, these can be obtained by applying the discrete Fourier transform to the point spread function array.

2.3.3. *Excursus*: Analysis of the Gaussian PSF

In this section we examine the Gaussian point spread function introduced in Section 2.2 with respect to its eigenvalues. From the previous section it is known that for

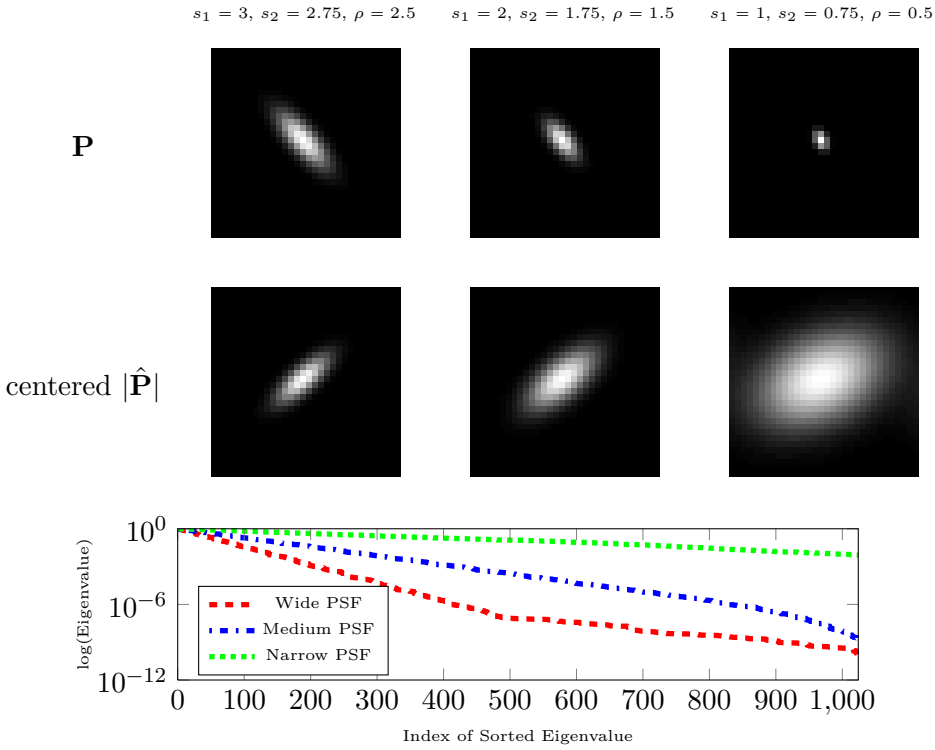


Figure 2.5.: Gaussian PSFs of different width and corresponding eigenvalues of $\mathbf{A}$ (image size: 32×32).

periodic boundary conditions

$$\begin{aligned} & \mathbf{b} = \mathbf{A}\mathbf{x} \\ \Leftrightarrow\ & \hat{\mathbf{b}} = \mathbf{\Lambda}\hat{\mathbf{x}} \\ \Leftrightarrow\ & \hat{\mathbf{b}} = \operatorname{diag}(\mathbf{F}\mathbf{p}^{\text{circ}})\hat{\mathbf{x}} \end{aligned} \tag{2.10}$$

that is the eigenvalues of the BCCB matrix $\mathbf{A}$ are given by the DFT of the circular shifted point spread function $\mathbf{P}^{\text{circ}}$. According to González and Woods [14, Ch.4], the spectrum of a Fourier transform is invariant to translation of the corresponding image. Hence, the spectrum of the Fourier transform of the PSF, $|\mathbf{F}\mathbf{p}|$, is equivalent to the spectrum of the circshifted Fourier transform given by $|\mathbf{F}\mathbf{p}^{\text{circ}}|$. Then, following (2.10), both spectra are equal to the absolute eigenvalues of $\mathbf{A}$ located on the diagonal of $|\mathbf{\Lambda}|$.

In Figure 2.5 we illustrate Gaussian point spread functions of different width and corresponding absolute eigenvalues. The first row shows the PSFs, opposed below to the centered spectra of their Fourier transforms. The centered view can be achieved by either multiplying the PSF $\mathbf{P}$ by $(-1)^{i+j}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ and subsequently performing the Fourier transform, or by applying a circular shift to the spectrum as described in the previous section. The comparison shows that the narrower the PSF, the wider the spectrum of its Fourier transform. This implies that

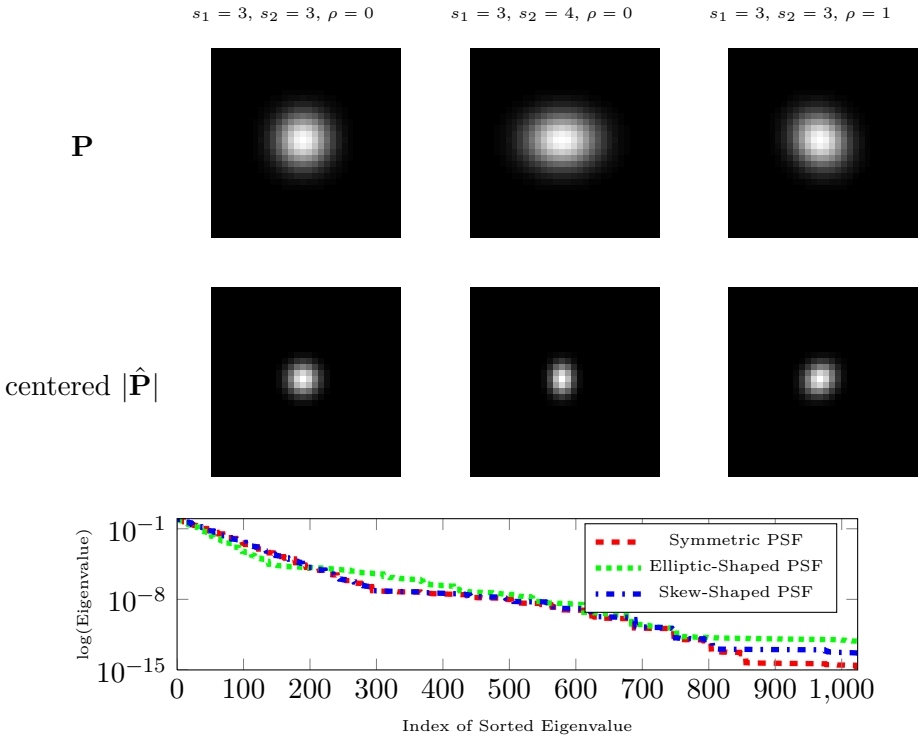


Figure 2.6.: Gaussian PSFs of similar width and corresponding eigenvalues of $\mathbf{A}$ (image size: 32×32).

the absolute eigenvalues of a narrow point spread function decrease slower than the eigenvalues of a wider PSF, which is also reflected in the plot at the bottom of Figure 2.5 comparing the absolute eigenvalues arranged in descending order. Moreover, one observes that the spectra are rotated by the same angle as $\mathbf{P}$, but in opposite direction, as explained in González and Woods [14, Ch.4].

On the other hand, Figure 2.6 shows a rotationally symmetric ($s_1 = s_2$, $\rho = 0$), an elliptic-shaped ($s_1 \neq s_2$, $\rho = 0$) and a skew-shaped ($s_1 \neq s_2$, $\rho \neq 0$) Gaussian point spread function of approximately the same width. Again, we illustrate the spectrum below each PSF, respectively. Here, we observe that the eigenvalues decrease with similar speed.

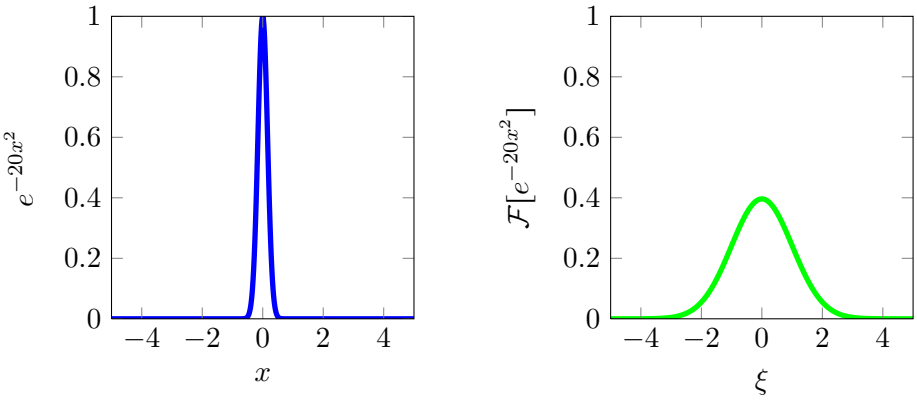
We explain that behavior regarding the one-dimensional case which may be interpreted as a line of intersection through the center of the two-dimensional PSF. The Fourier transform of a one-dimensional Gaussian function $f(x) = e^{-ax^2}$,

$$\mathcal{F}[e^{-ax^2}](\xi) = \sqrt{\frac{\pi}{a}} e^{\frac{-\pi^2\xi^2}{a}},$$

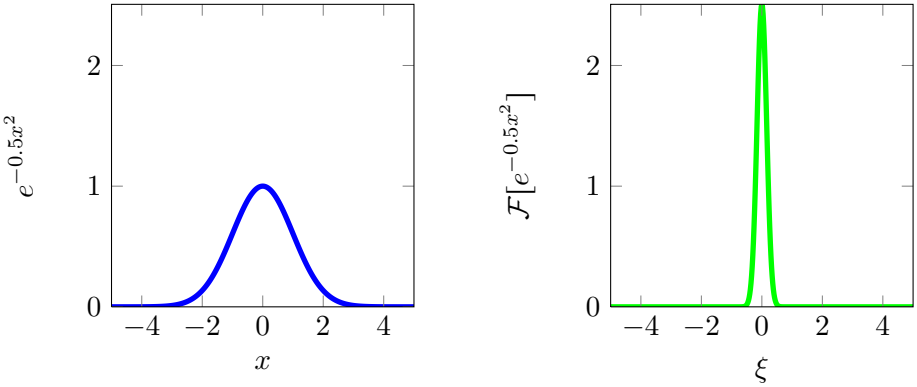
is a Gaussian function as well.

The larger we choose parameter a, the narrower the Gaussian and the wider its Fourier

transform. We give an example to demonstrate that fact in Figure 2.7.



(a) Narrow Gaussian function (*left*) and corresponding Fourier transform (*right*) with $a = 20$.



(b) Wide Gaussian function (*left*) and corresponding Fourier transform (*right*) with $a = 0.5$.

Figure 2.7.: Comparison of different one-dimensional Gaussian functions and Fourier transforms.

The same holds for higher dimensional Gaussian functions. This implies that for every Gaussian PSF, the Fourier transform is a Gaussian PSF of different width, and thus positive and real. Therefore, all eigenvalues of $\mathbf{A}$ are positive and real. Summing up, we can assume that the eigenvalues of a narrow Gaussian PSF decrease slower than the eigenvalues of a wider Gaussian PSF.

Moreover, further experiments with Gaussian PSFs indicate that for the eigenvalues as they appear in $\mathbf{\Lambda} = \text{diag}(\boldsymbol{\lambda})$ where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{mn})^T$, the absolute value of eigenvalue λ_i of the narrowest PSF is larger than the absolute value of eigenvalue λ_i of less narrow PSFs for all $i = 1, \ldots, mn$. This is illustrated in Figure 2.8 which corresponds to the example in Figure 2.5.

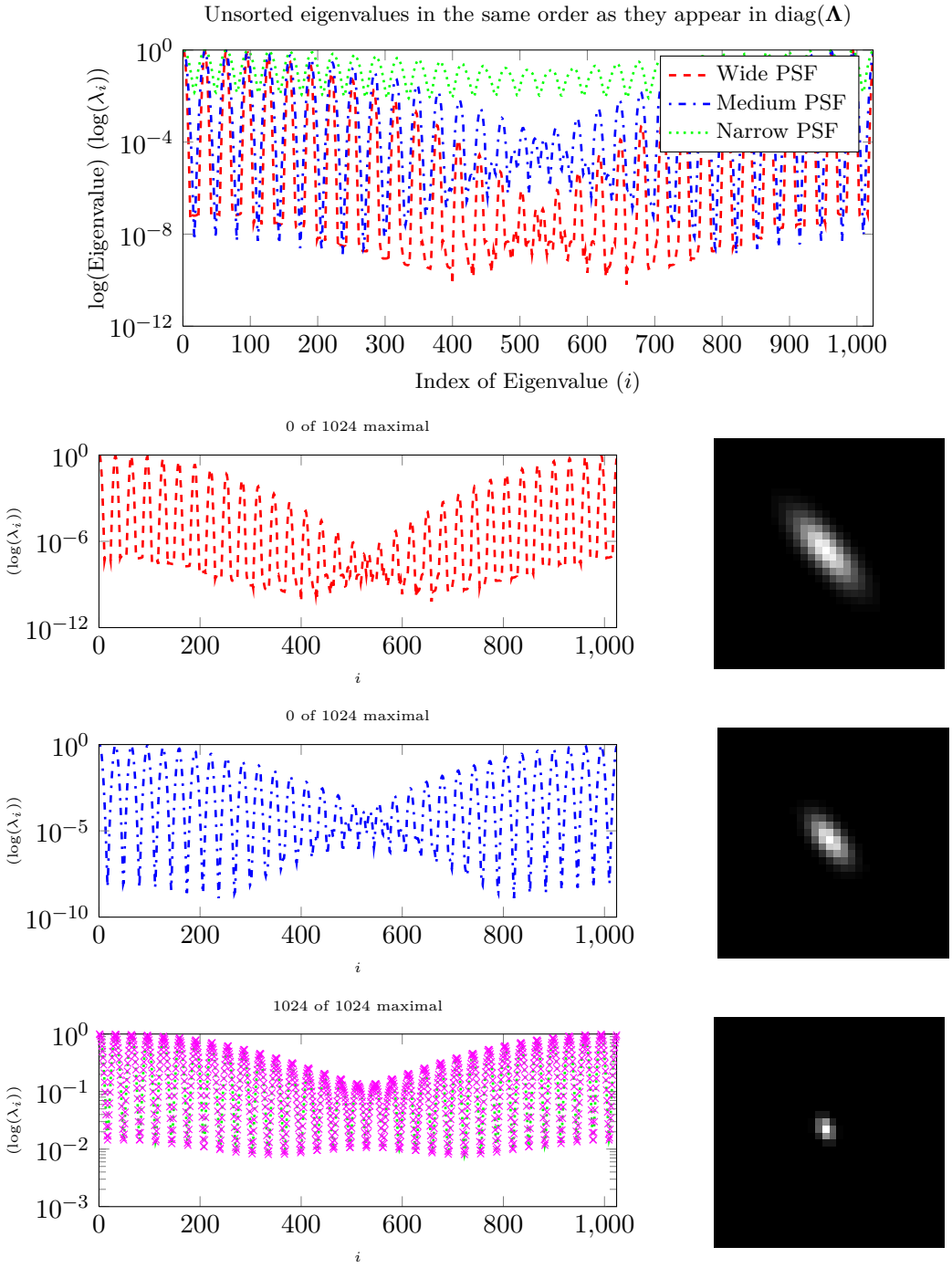


Figure 2.8.: *Top*: Absolute eigenvalues in the same order as they appear in $\mathbf{\Lambda}$ corresponding to the three exemplary PSFs in Figure 2.5. *Bottom*: Identification of the PSF corresponding to the maximum absolute eigenvalue at position $i = 1, \ldots, mn$ marked with a magenta cross, respectively. One observes that the narrowest PSF contains the componentwise maximum absolute eigenvalues.

(a) X **(b) B** (zero)

(c) B (periodic) **(d) B** (reflexive)

Figure 2.9.: True image **X** and corresponding blurred images using zero, periodic and reflexive boundary conditions, respectively (image size: 512×512, blur: Gaussian with $s_1 = s_2 = 10, \rho = 0$).

2.3.4. Reflexive Boundary Conditions

Regarding the satellite image we used in our illustrations above, zero, periodic and reflexive boundary conditions can be applied due to its black background. On the contrary, Figure 2.9 illustrates an image where modelling of the blurring process using reflexive boundary conditions is advisable, since zero and periodic extension results in improper pixel values or artifacts close to the borders of the blurred image.

In the reflexive case, the blurring matrix **A** consists of the sum of four matrices,

- a block Toeplitz matrix with Toeplitz blocks (BTTB),
- a block Toeplitz matrix with Hankel blocks (BTHB),
- a block Hankel matrix with Toeplitz blocks (BHTB),
- and a block Hankel matrix with Hankel blocks (BHHB).

The format of a Toeplitz matrix is described in (2.5). In contrast, an $N \times N$ Hankel

matrix [19] exhibits the structure

$$\mathbf{H} = \begin{pmatrix} h_0 & h_1 & h_2 & \dots & h_{N-1} \\ h_1 & h_2 & h_3 & \dots & h_N \\ h_2 & h_3 & \ddots & & h_{N+1} \\ \vdots & \vdots & & \ddots & \vdots \\ h_{N-1} & h_N & h_{N+1} & \dots & h_{2(N-1)} \end{pmatrix}.$$

The blurring process of a 3×3 image employing reflexive boundary conditions can be described by

$$\mathbf{b} = [\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_4]\mathbf{x},$$

where $\mathbf{x}, \mathbf{b} \in \mathbb{R}_+^{9\times 1}$ and blurring matrix $\mathbf{A} \in \mathbb{R}_+^{9\times 9}$ is composed of the sum of

$$\mathbf{A}_1 = \left(\begin{array}{ccc|ccc|ccc} p_{22} & p_{21} & & p_{12} & p_{11} & & & & \\ p_{23} & p_{22} & p_{21} & p_{13} & p_{12} & p_{11} & & & \\ & p_{23} & p_{22} & & p_{13} & p_{12} & & & \\ \hline p_{32} & p_{31} & & p_{22} & p_{21} & & p_{12} & p_{11} & \\ p_{33} & p_{32} & p_{31} & p_{23} & p_{22} & p_{21} & p_{13} & p_{12} & p_{11} \\ & p_{33} & p_{32} & & p_{23} & p_{22} & & p_{13} & p_{12} \\ \hline & & & p_{32} & p_{31} & & p_{22} & p_{21} & \\ & & & p_{33} & p_{32} & p_{31} & p_{23} & p_{22} & p_{21} \\ & & & & p_{33} & p_{32} & & p_{23} & p_{22} \end{array}\right)$$

$$\mathbf{A}_2 = \left(\begin{array}{ccc|ccc|ccc} p_{23} & & & p_{13} & & & & & \\ & & p_{21} & & & p_{11} & & & \\ \hline p_{33} & & & p_{23} & & & p_{13} & & \\ & & p_{31} & & & p_{21} & & & p_{11} \\ \hline & & & p_{33} & & & p_{23} & & \\ & & & & & p_{31} & & & p_{21} \end{array}\right)$$

$$\mathbf{A}_3 = \left(\begin{array}{ccc|c|cc} p_{32} & p_{31} & & & & \\ p_{33} & p_{32} & p_{31} & & & \\ & p_{33} & p_{32} & & & \\ \hline & & & & & \\ & & & & & \\ & & & & & \\ \hline & & & & p_{12} & p_{11} \\ & & & & p_{13} & p_{12} \\ & & & & & p_{13} \end{array}\right)$$

$$\mathbf{A}_4 = \left(\begin{array}{ccc|c|cc} p_{33} & & & & & \\ & & p_{31} & & & \\ \hline & & & & & \\ & & & & & \\ & & & & & \\ \hline & & & & p_{13} & \\ & & & & & \\ & & & & & p_{11} \end{array}\right)$$

which are BTTB, BTHB, BHTB and BHHB matrices, respectively.

The spectral decomposition of $\mathbf{A}$ can be computed very efficiently using the discrete cosine transform if the underlying PSF is doubly symmetric. Imagining two lines, one drawn horizontally, the other vertically, through the center of the two-dimensional PSF image, the PSF is called doubly symmetric if both lines are symmetry axes. For instance, these demands are met by a Gaussian PSF with s_1 and s_2 arbitrarily chosen and $\rho = 0$. The diagonalization is similar to applying periodic boundary conditions and is given by

$$\mathbf{A} = \mathbf{K}^T \mathbf{\Lambda} \mathbf{K},$$

where $\mathbf{\Lambda}$ is, as defined in (2.8), a diagonal matrix containing the eigenvalues of $\mathbf{A}$ and $\mathbf{K}$ describes the two-dimensional DCT (discrete cosine transform) matrix. For more details concerning reflexive boundary conditions, see the book by Hansen et al. [16].

Supposing that an image is repeated reflexively beyond its borders is, in many cases, more realistic than inducing zero or periodic boundary conditions. On the other hand, efficient implementation of the method is limited due to symmetry constraints regarding the PSF.

In the following, we limit our studies to periodic boundary conditions. This is reasonable, since due to the spectral decomposition, fast computations can be performed in the Fourier domain. Moreover, there are no restrictions regarding the PSF, as is the case for reflexive boundary conditions. Given these facts, we accept little artifacts based on unrealistic continuation beyond the image's borders.

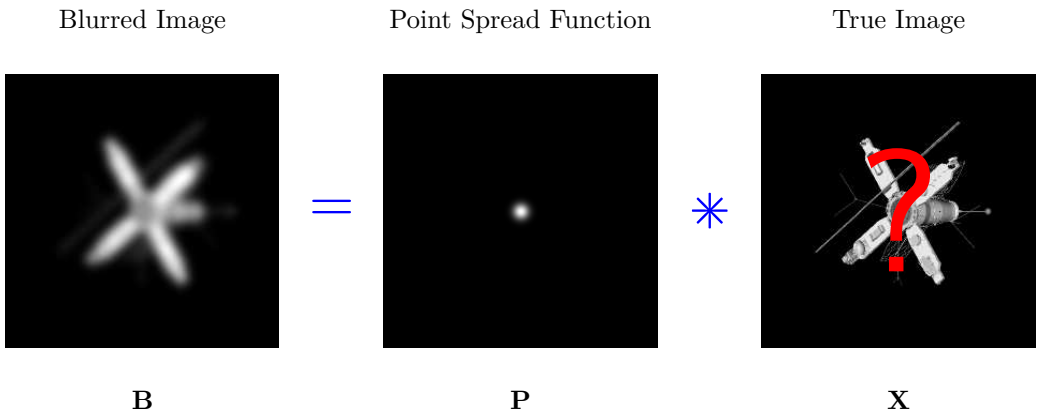


Figure 2.10.: The classic deconvolution problem.

2.4. Solving the Deblurring Problem

In the previous section we described the blurring process as a matrix-vector multiplication $\mathbf{Ax} = \mathbf{b}$ with true image $\mathbf{x}$, blurred image $\mathbf{b}$ and blurring matrix $\mathbf{A}$. Next, we concentrate on solving the deblurring problem illustrated in Figure 2.10, that is to determine the sharp image $\mathbf{x}$ given $\mathbf{A}$ and $\mathbf{b}$. This challenge is called a linear *inverse problem*. The solution can be explicitly stated as $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ if matrix $\mathbf{A}$ is square and nonsingular. However, this formula does not yield the true image $\mathbf{x}$ but a version that involves errors due to ill-conditioning of $\mathbf{A}$.

2.4.1. Noise and Errors

In addition to undesired effects caused by an ill-conditioned matrix $\mathbf{A}$, the recording process has an impact on the solution. Due to the fact that a picture is taken by a mechanical device, we have to assume that the blurred image is contaminated by a certain amount of noise. Moreover, image quantization leads to inaccuracies. We model these perturbations as additive noise

$$\mathbf{Ax} = \mathbf{b} = \mathbf{b}_{\text{exact}} + \mathbf{e},$$

where $\mathbf{e} \in \mathbb{R}^{mn \times 1}$. The solution

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}_{\text{exact}} + \mathbf{A}^{-1}\mathbf{e}$$

indicates that the exact solution is influenced by a term $\mathbf{A}^{-1}\mathbf{e}$ called *inverted noise*. The larger the *inverted noise* part compared to the exact solution, the more inaccurate the reconstruction. The impact of errors and noise on the solution can be decreased using *regularization* methods.

2.4.2. Tikhonov Regularization Using Generalized Cross-Validation

As introduced in Section 2.4.1, solving the optimization problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - (\mathbf{b}_{\text{exact}} + \mathbf{e})\|_2^2$$

is difficult, because simply minimizing the expression above yields a solution influenced by noise. Regularization seeks to find a balance between a good least squares fit and a solution containing a small inverted noise component. A simple but efficient regularization method that we examine below is called *Tikhonov* regularization [11].

First, consider the singular value decomposition (SVD) of a nonsingular PSF matrix $\mathbf{A} \in \mathbb{R}_+^{N\times N}$,

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

where $\mathbf{U} \in \mathbb{R}^{N\times N}$ and $\mathbf{V} \in \mathbb{R}^{N\times N}$ are orthogonal matrices and $\mathbf{\Sigma} \in \mathbb{R}^{N\times N}$ is a diagonal matrix containing the singular values σ_i, $i = 1, \ldots, N$ of $\mathbf{A}$ in decreasing order $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_N > 0$. Hence, a naive reconstruction, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, can be written as

$$\mathbf{x} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{b} = \sum_{i=1}^{N} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$, $i = 1, \ldots, N$ define the columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. As shown by Hansen et al. [16], the $|\mathbf{u}_i^T\mathbf{b}|$ decay until a *noise plateau* is reached, where $\mathbf{u}_i^T\mathbf{b} \approx \mathbf{u}_i^T\mathbf{b}_{\text{exact}}$ above and $\mathbf{u}_i^T\mathbf{b} \approx \mathbf{u}_i^T\mathbf{e}$ below this level. Thus very small singular values σ_i correspond to the part of $\mathbf{b}$ that is contaminated by noise, and $\mathbf{x}$ becomes large when it contains a large amount of inverted noise.

Using Tikhonov regularization we constrain the size of $\mathbf{x}$ by adding the weighted norm of $\mathbf{x}$, as a so called regularization term to the optimization problem. The Tikhonov regularized expression of the minimization problem is then given by

$$\min_{\mathbf{x}} \left\{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \epsilon^2\|\mathbf{x}\|_2^2\right\} \tag{2.11}$$

with *regularization parameter* $\epsilon > 0$. Expression (2.11) is equivalent to the form

$$\min_{\mathbf{x}} \left\{ \left\| \begin{pmatrix} \mathbf{A} \\ \epsilon\mathbf{I}_N \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\|_2^2 \right\} \tag{2.12}$$

which allows us to choose one of numerous numerical methods for least squares problems to solve the regularized problem.

A difficult problem, however, is to select a regularization parameter ϵ that adequately weights the regularization term. In the following, we describe an often used method to find the regularization parameter called *generalized cross-validation* [13; 16].

The generalized cross-validation (GCV) functional is defined by

$$G(\epsilon) = \frac{\|(\mathbf{I}_N - \mathbf{A}\mathbf{V}\mathbf{\Phi}\mathbf{\Sigma}^{-1}\mathbf{U}^T)\mathbf{b}\|_2^2}{(\mathrm{tr}(\mathbf{I}_N - \mathbf{A}\mathbf{V}\mathbf{\Phi}\mathbf{\Sigma}^{-1}\mathbf{U}^T))^2}, \tag{2.13}$$

where $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is the identity matrix. Moreover, $\mathrm{tr}(\cdot)$ denotes the trace, that is the sum of the diagonal elements of a square matrix [19, p.40], and matrix $\mathbf{\Phi} \in \mathbb{R}^{N\times N}$ is a diagonal matrix given by

$$\mathbf{\Phi} = \mathrm{diag}\,(\boldsymbol{\phi}), \quad \boldsymbol{\phi} = \left(\frac{\sigma_1^2}{\sigma_1^2+\epsilon^2}, \ldots, \frac{\sigma_N^2}{\sigma_N^2+\epsilon^2}, 0, \ldots, 0\right)^T$$

with singular values σ_i, $i = 1, \ldots, N$ of matrix $\mathbf{A}$. The perfect parameter with respect to the GCV method is an $\epsilon > 0$ that minimizes (2.13). In the book by Hansen et al. [16], it is described how the minimum of the GCV functional $G(\epsilon)$ can be efficiently determined. Algebraic simplifications lead to an expression whose minimum can be found using the MATLAB function `fminbnd`. Similarly, the GCV functional can be set up using spectral decomposition instead of SVD.

Now that we have described a regularization and a parameter choice method, we turn our attention to finally solving the deblurring problem. We illustrate one possible method to compute $\mathbf{x}$ using *normal equations* below.

2.4.3. The Method of Normal Equations

Again, we consider the problem of calculating a vector $\mathbf{x}$ that minimizes

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \tag{2.14}$$

where matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$, $M \geqslant N$, is not necessarily square. The *method of normal equations* is a simple but fast approach to solve such a least squares problem for $\mathbf{x}$. Expression (2.14) can equally be stated as

$$(\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b}). \tag{2.15}$$

To optimize, we differentiate (2.15) with respect to $\mathbf{x}$ and set the result equal to zero. These steps yield equation

$$\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b} = 0$$

whose rearranged form

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

is called the *normal equation*. Let $\mathbf{A}^T\mathbf{A} = \mathbf{M} \in \mathbb{R}^{N\times N}$ and $\mathbf{A}^T\mathbf{b} = \mathbf{y} \in \mathbb{R}^{N\times 1}$. The advantage of formulating the normal equation is that it leads to a system of equations $\mathbf{M}\mathbf{x} = \mathbf{y}$ with a square and in many cases much smaller matrix. On the other hand, if

$\mathbf{A}$ is ill-conditioned, the condition number is squared in matrix $\mathbf{M}$ leading to numerical difficulties [12, p.239].

In particular, applying the method of normal equations to the Tikhonov regularized optimization problem in (2.12) we receive the equation

$$(\mathbf{A}^T\mathbf{A} + \epsilon^2\mathbf{I}_N)\mathbf{x} = \mathbf{A}^T\mathbf{b}. \tag{2.16}$$

However, for many applications in image processing $\mathbf{A}$ is a dense matrix and the product $\mathbf{A}^T\mathbf{A}$ still large. Hence, solving (2.16) for $\mathbf{x}$ is computationally expensive.

Assuming boundary conditions to be periodic or reflexive, as described in Sections 2.3.2 and 2.3.4 respectively, we can reduce computational complexity of the image deblurring problem. For instance, in the periodic case the spectral decomposition of $\mathbf{A} \in \mathbb{R}_+^{N\times N}$ consists of DFT matrices and diagonal matrix as given in (2.7). Thus, the normal equation can be written in Fourier representation as

$$\underbrace{(\mathbf{\Lambda}^*\mathbf{\Lambda} + \epsilon^2\mathbf{I}_N)}_{\hat{\mathbf{M}}_\epsilon}\hat{\mathbf{x}} = \mathbf{\Lambda}^*\hat{\mathbf{b}}, \tag{2.17}$$

where diagonal matrix $\mathbf{\Lambda}^*$ contains the complex conjugates of the eigenvalues of $\mathbf{A}$. Since overall matrix $\hat{\mathbf{M}}_\epsilon$ is also a diagonal matrix, computations can be performed very efficiently. Similarly, in the case of reflexive boundary conditions, we can compute $\mathbf{x}$ using the DCT, provided the PSF satisfies the required symmetry conditions.

Algorithm 2.1 summarizes the regularized reconstruction of $\mathbf{x}$ using the Tikhonov method with GCV.

Tikhonov Method and GCV

```
1  function ϵ = GCV(A,b)
2
3  ϵ = argmin(G(ϵ))                          % G(ϵ) as defined in (2.13)
4
5  function x = Tikhonov(A,b,ϵ)
6
7  x = (A^T A+ϵ^2 I_N)^{-1} A^T b
```

Algorithm 2.1

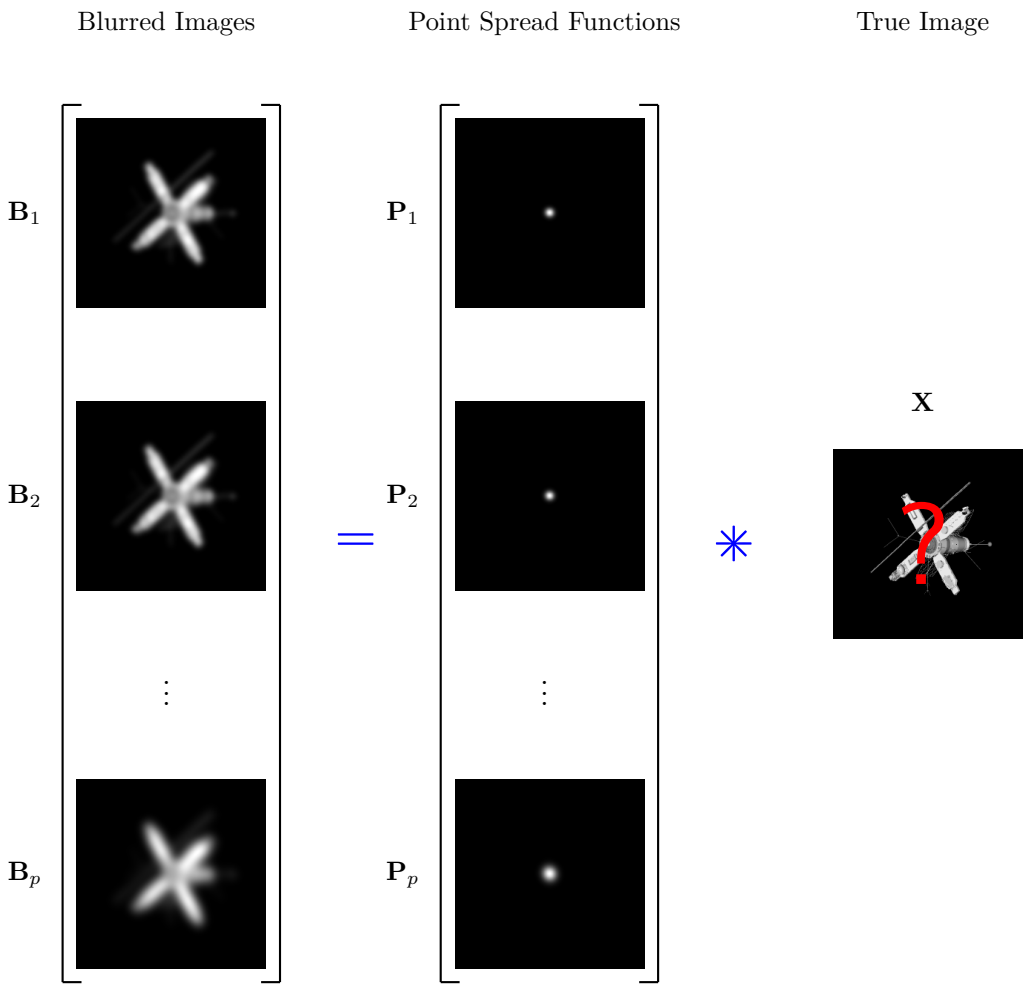


Figure 2.11.: The multiframe deconvolution problem.

2.5. Multiframe Image Deblurring

Multiframe image deblurring expands the image deblurring challenge from (2.4) to a problem where we try to reconstruct an image given not only one but multiple variably blurred images of the same object. Figure 2.11 illustrates the multiframe blurring problem for p different frames. Various applications, especially in astronomy, make this an important problem in image processing. For instance, multiframe deconvolution is a useful postprocessing tool for data obtained by the Large Binocular Telescope (LBT). Compared to classic deconvolution that only uses one observed image, simultaneously deconvolving images taken with different orientation of the LBT can improve spatial resolution of the reconstructed object (Bertero and Boccacci [2]).

We can formulate multiframe image deblurring as an extremely overdetermined least

squares problem

$$\min_{\mathbf{x}} \sum_{j=1}^{p} \|\mathbf{A}_j\mathbf{x} - \mathbf{b}_j\|_2^2 = \min_{\mathbf{x}} \left\| \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_p \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_p \end{pmatrix} \right\|_2^2, \tag{2.18}$$

where the true image is given by $\mathbf{x} \in \mathbb{R}_+^{mn\times 1}$ and the multiple blurred images, corresponding to the blur matrices $\mathbf{A}_1, \ldots, \mathbf{A}_p \in \mathbb{R}_+^{mn\times mn}$, are denoted by $\mathbf{b}_1, \ldots, \mathbf{b}_p \in \mathbb{R}_+^{mn\times 1}$.

In the following, we induce periodic boundary conditions in order to substitute $\mathbf{A}_j$ with its spectral decomposition $\mathbf{F}^*\mathbf{\Lambda}_j\mathbf{F}$ (see (2.7)). As a result, the optimization problem can be restated as

$$\min_{\hat{\mathbf{x}}} \left\| \begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix} \hat{\mathbf{x}} - \begin{pmatrix} \hat{\mathbf{b}}_1 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix} \right\|_2^2 \tag{2.19}$$

where $\mathbf{\Lambda}_j = \operatorname{diag}\left(\boldsymbol{\lambda}^{(j)}\right), \quad \boldsymbol{\lambda}^{(j)} = \left(\lambda_1^{(j)}, \ldots, \lambda_{mn}^{(j)}\right)^T$. The expressions $\hat{\mathbf{x}}$ and $\left(\hat{\mathbf{b}}_1; \ldots; \hat{\mathbf{b}}_p\right)$ describe the Fourier transform of true and blurred images respectively.

Compared to (2.18), formulation (2.19) simplifies the problem. Instead of full matrices $\mathbf{A}_j$ the matrix $(\mathbf{\Lambda}_1; \ldots; \mathbf{\Lambda}_p) \in \mathbb{C}^{mnp\times mn}$ is sparse, consisting only of blocks of diagonal matrices.

Thus, the normal equation of (2.19) can be written in Fourier representation as

$$\sum_{j=1}^{p} \mathbf{\Lambda}_j^* \mathbf{\Lambda}_j \hat{\mathbf{x}} = \sum_{j=1}^{p} \mathbf{\Lambda}_j^* \hat{\mathbf{b}}_j$$

where diagonal matrix $\mathbf{\Lambda}_j^*$ contains the complex conjugates of the eigenvalues of $\mathbf{A}$. Equivalently, this expression can be stated as a single matrix-vector multiplication

$$\underbrace{\begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix}^* \begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix}}_{\hat{\mathbf{M}}^{(1:p)}} \hat{\mathbf{x}} = \begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix}^* \begin{pmatrix} \hat{\mathbf{b}}_1 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix} \tag{2.20}$$

where matrix-matrix product $\hat{\mathbf{M}}^{(1:p)} \in \mathbb{R}^{mn\times mn}$ is a diagonal matrix. The ith entry of the diagonal is given by

$$\hat{m}_{ii}^{(1:p)} = \sum_{j=1}^{p} |\lambda_i^{(j)}|^2$$

where $i = 1, \ldots, mn$. Due to the fact that some eigenvalues are very close to zero, matrix $\hat{\mathbf{M}}^{(1:p)}$ is ill-conditioned leading to numerical errors in the direct solution that we obtain by rearranging equation (2.20) for $\hat{\mathbf{x}}$. To receive a reasonable solution we have

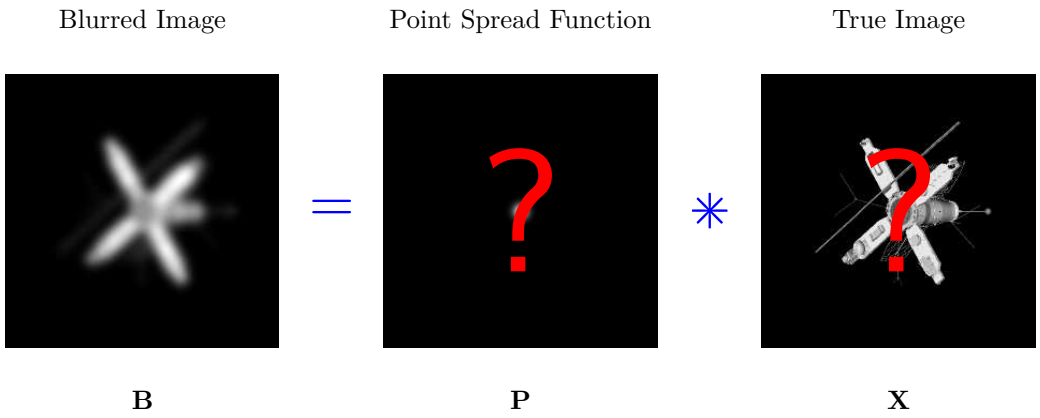


Figure 2.12.: The blind deconvolution problem.

to introduce regularization such as the previously described Tikhonov regularization. According to (2.17), the Tikhonov regularized expression of the Fourier representation is given by

$$\underbrace{\left(\hat{\mathbf{M}}^{(1:p)} + \epsilon^2 \mathbf{I}_{mn}\right)}_{\hat{\mathbf{M}}_\epsilon^{(1:p)}} \hat{\mathbf{x}} = \begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix}^* \begin{pmatrix} \hat{\mathbf{b}}_1 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix}.$$

with diagonal matrix $\hat{\mathbf{M}}_\epsilon^{(1:p)}$.

2.6. Blind Image Deblurring

In a general blind deconvolution problem as demonstrated in Figure 2.12, only the blurred image is known and we want to reconstruct the true image and PSF. In particular, we have to solve the underdetermined optimization problem

$$\min_{\mathbf{x},\mathbf{y}} \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2. \tag{2.21}$$

with true image $\mathbf{x} \in \mathbb{R}_+^{mn\times 1}$, blurred image $\mathbf{b} \in \mathbb{R}_+^{mn\times 1}$, and PSF matrix $\mathbf{A} \in \mathbb{R}_+^{mn\times mn}$ non-linearly depending on a certain number of parameters that are collected in vector $\mathbf{y}$. Although the type of the PSF, for instance Gaussian, is known, the parameters that precisely define the PSF, like $\mathbf{y} = [s_1; s_2; \rho]$ for a Gaussian PSF, are unknown.

This is a complicated, ill-posed and ill-conditioned problem. As for the classic image deconvolution and the multiframe problem, using regularization, here the Tikhonov method stated above, is crucial to obtain a satisfactory reconstruction.

According to Kundur and Hatzinakos [22] there are two different approaches to solve a blind image deconvolution problem:

1. Determine PSF matrix $\mathbf{A}$ separately (prior to image restoration). Subsequently, use classical deblurring approaches to obtain an approximation of $\mathbf{x}$.

2. Simultaneously reconstruct PSF matrix $\mathbf{A}$ and true image $\mathbf{x}$.

In Section 3.1 we examine an approach that falls in the first category where the PSF is determined using the so called *APEX method* [6]. Methods belonging to the second category are discussed in Section 3.2.

3. Approaches for Blind Deconvolution

In the following we analyze two different approaches to solve the blind deconvolution problem (2.21) introduced in the previous section. First, we consider a method to separately compute the point spread function and deblur the image. Moreover, we examine ways to simultaneously reconstruct the PSF and true image using iterative methods.

3.1. Separate Computation of PSF and Reconstructed Image

One option to tackle the blind deconvolution problem is to first apply a method that determines the PSF. This results in a classic deconvolution problem as given in (2.4) where the blurred image and cause of the blur are known and we seek to reconstruct sharp image $\mathbf{x}$. We illustrate this stepwise approach in Figure 3.1. Below, we examine an interesting technique, the APEX method [6], that can be used to complete the first step.

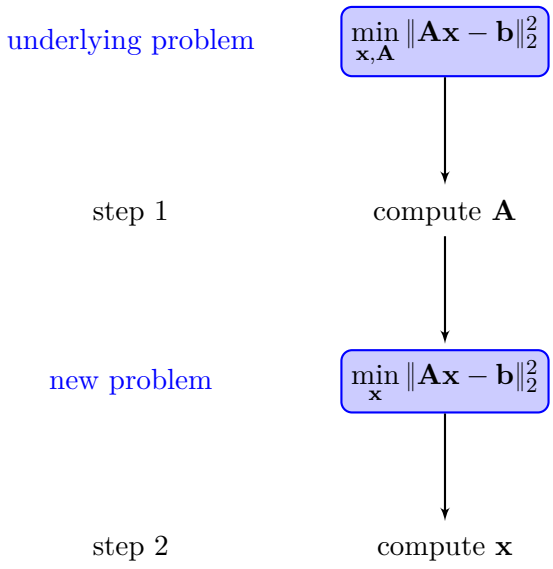


Figure 3.1.: Direct model for solving the blind deconvolution problem. $\mathbf{A}$ and $\mathbf{x}$ are computed separately.

3.1.1. APEX Method

In Carasso [6] an algorithm for blind deconvolution, the so called APEX method, has been developed to estimate the PSF using information of the blurred image. For this purpose, the PSF has to be symmetric Lévy stable. We define a symmetric Lévy stable PSF as a function whose Fourier transform is given by

$$\begin{aligned}\hat{P}_{\alpha,\beta} : \mathbb{R}^2 \to \mathbb{R}_+, \quad \hat{P}_{\alpha,\beta}(\xi,\eta) &= \int_{\mathbb{R}^2} P(s,t) \exp\left\{-2\pi i(s\xi + t\eta)\right\} \\ &= \exp\left\{-\alpha\left(\xi^2+\eta^2\right)^{\beta}\right\} \end{aligned} \tag{3.1}$$

where $\alpha > 0$ and $0 < \beta \leqslant 1$. For instance, this group includes Gaussian PSFs for $\beta = 1$, Lorentzian PSFs for $\beta = \frac{1}{2}$ and long-exposure atmospheric turbulence blur for $\beta = \frac{5}{6}$.

It is known from above, that we can express the blurring process by a matrix-vector multiplication $\mathbf{b} = \mathbf{A}\mathbf{x}$, and by applying periodic boundary conditions the equation can be rewritten as $\hat{\mathbf{b}} = \mathbf{\Lambda}\hat{\mathbf{x}}$ where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{A}$. Moreover, we previously described that the diagonal of $\mathbf{\Lambda}$ containing the eigenvalues is equal to the vector representation of the two-dimensional DFT of the circshifted point spread function corresponding to matrix $\mathbf{A}$.

We define an array representation of expression (3.1) for symmetric Lévy stable PSFs by

$$\hat{\mathbf{P}} = \exp\left\{-\alpha\left(\mathbf{\Xi}^2 + \mathbf{H}^2\right)^{\beta}\right\}, \tag{3.2}$$

where

$$\begin{aligned}\mathbf{\Xi} &= \mathbf{1}_m \otimes \boldsymbol{\xi} \quad \in \mathbb{N}^{m\times n} \\ \mathbf{H} &= \boldsymbol{\eta} \otimes \mathbf{1}_n^T \quad \in \mathbb{N}^{m\times n}\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{\xi} &= \left(-\left\lceil\frac{n}{2}\right\rceil + 1, -\left\lceil\frac{n}{2}\right\rceil + 2, \ldots, 0, \ldots, \left\lfloor\frac{n}{2}\right\rfloor - 1, \left\lfloor\frac{n}{2}\right\rfloor\right) \quad \in \mathbb{N}^{1\times n} \\ \boldsymbol{\eta} &= \left(-\left\lceil\frac{m}{2}\right\rceil + 1, -\left\lceil\frac{m}{2}\right\rceil + 2, \ldots, 0, \ldots, \left\lfloor\frac{m}{2}\right\rfloor - 1, \left\lfloor\frac{m}{2}\right\rfloor\right) \quad \in \mathbb{N}^{1\times m}.\end{aligned}$$

Note that the exponential function and squaring are applied elementwise.

Since the Fourier transformed PSF $\hat{P}$ stated in (3.1) is real and positive, its discretized form $\hat{\mathbf{P}}$ has only real and positive entries, as well. Therefore, $\hat{\mathbf{P}}$ and the Fourier transform of circshifted PSF, $\hat{\mathbf{P}}^{\text{circ}}$, are the same and we can substitute $\mathbf{\Lambda}$ by a pointwise multiplication by $\hat{\mathbf{P}}$

$$\hat{\mathbf{B}} = \hat{\mathbf{P}} \odot \hat{\mathbf{X}} \tag{3.3}$$

where $\mathbf{B}$, $\mathbf{P}$ and $\mathbf{X}$ are $m \times n$ sized blurred image, point spread function and true image, respectively, and $\odot$ denotes the elementwise multiplication of two arrays.

In Chapter 2, Section 2.3.2, we regarded the structure of the DFT matrix $\mathbf{F}$. We discovered, that $\mathbf{F}$ has one row and one column consisting of ones, and all other entries are located in the interval $[-1, 1[$. Since $\mathbf{X}$, $\mathbf{B}$, and $\mathbf{P}$ are supposed to have only non-negative entries, the maximum entry of $\hat{\mathbf{X}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{P}}$ results from multiplying the row of ones with vectors $\mathbf{x}$, $\mathbf{b}$, and $\mathbf{p}$, respectively. Hence, the maximum value of the Fourier transforms of the true image, blurred image, and the PSF is located at the same array position. Moreover, it is known from Chapter 2, Section 2.2, that the sum of all entries of the PSF array is equal to one, which coincides with the maximum value of $\hat{\mathbf{P}}$. Thus, according to (3.3), the maximum value of $\hat{\mathbf{X}}$ and $\hat{\mathbf{B}}$ is the same.

Dividing equation (3.3) by $\gamma = \max(\hat{\mathbf{B}}) = \max(\hat{\mathbf{X}})$

$$\frac{\hat{\mathbf{B}}}{\gamma} = \hat{\mathbf{P}} \odot \frac{\hat{\mathbf{X}}}{\gamma} \Leftrightarrow \hat{\mathbf{B}}^{\gamma} = \hat{\mathbf{P}} \odot \hat{\mathbf{X}}^{\gamma} \tag{3.4}$$

we define by $\hat{\mathbf{B}}^{\gamma} := \frac{\hat{\mathbf{B}}}{\gamma}$ and $\hat{\mathbf{X}}^{\gamma} := \frac{\hat{\mathbf{X}}}{\gamma}$ the normalized Fourier transforms of blurred and true image. Taking the logarithm of (3.4) yields equation

$$\log|\hat{\mathbf{B}}^{\gamma}| = \log|\hat{\mathbf{P}}| + \log|\hat{\mathbf{X}}^{\gamma}| \tag{3.5}$$

denoting by $\log|\hat{\mathbf{B}}^{\gamma}|$, $\log|\hat{\mathbf{P}}|$, and $\log|\hat{\mathbf{X}}^{\gamma}|$ the elementwise absolute and subsequently logarithmized pixel values of $\hat{\mathbf{B}}^{\gamma}$, $\hat{\mathbf{P}}$, and $\hat{\mathbf{X}}^{\gamma}$, respectively.

Then, substituting $\hat{\mathbf{P}}$ by array representation of the expression for symmetric Lévy stable PSFs stated in (3.2) the equation describing the blurring process can be converted to

$$\log|\hat{\mathbf{B}}^{\gamma}| = -\alpha\left(\boldsymbol{\Xi}^2 + \mathbf{H}^2\right)^{\beta} + \log|\hat{\mathbf{X}}^{\gamma}|, \tag{3.6}$$

where all operations are componentwise, as well.

Next, we consider $\log|\hat{\mathbf{B}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)|$, the row through the center of array $\log|\hat{\mathbf{B}}^{\gamma}|$. According to (3.6), the middle row can be stated as

$$\begin{aligned}&\log\left|\hat{\mathbf{B}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)\right| = \\ &-\alpha\left(\boldsymbol{\Xi}^2\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right) + \mathbf{H}^2\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)\right)^{\beta} + \log\left|\hat{\mathbf{X}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)\right|\end{aligned}$$

which simplifies to

$$\log\left|\hat{\mathbf{B}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)\right| = -\alpha\boldsymbol{\Xi}^{2\beta}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right) + \log\left|\hat{\mathbf{X}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)\right|, \tag{3.7}$$

since row $\left\lfloor\frac{m}{2}\right\rfloor$ of $\mathbf{H}$ contains only zero values.

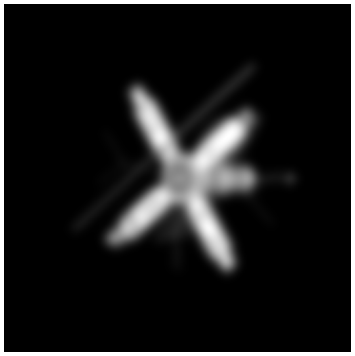
Plotting the entries of $\log|\hat{\mathbf{B}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)|$ against the n equidistant values

$$-\left\lceil\frac{n}{2}\right\rceil + 1, -\left\lceil\frac{n}{2}\right\rceil + 2, \ldots, 0, \ldots, \left\lfloor\frac{n}{2}\right\rfloor - 1, \left\lfloor\frac{n}{2}\right\rfloor,$$

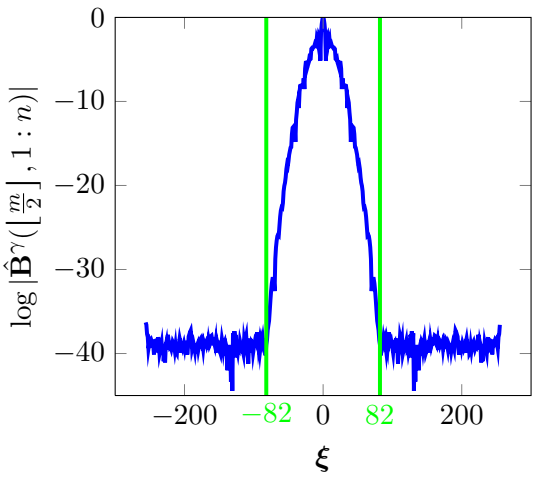
as illustrated in Figure 3.2 (b) for the blurred image in Figure 3.2 (a), we detect the interval $[-\omega, \omega]$, with $\mathbb{N} \ni \omega \leqslant \left\lfloor \frac{n}{2} \right\rfloor$, for which $\log |\hat{\mathbf{B}}^{\gamma} \left(\left\lfloor \frac{m}{2} \right\rfloor, 1 : n \right)|$ lies above the noise level, and define a vector

$$\boldsymbol{\xi}_{\omega} = (-\omega, -\omega + 1, \ldots, 0, \ldots, \omega - 1, \omega) \in \mathbb{N}^{1 \times (2\omega+1)}$$

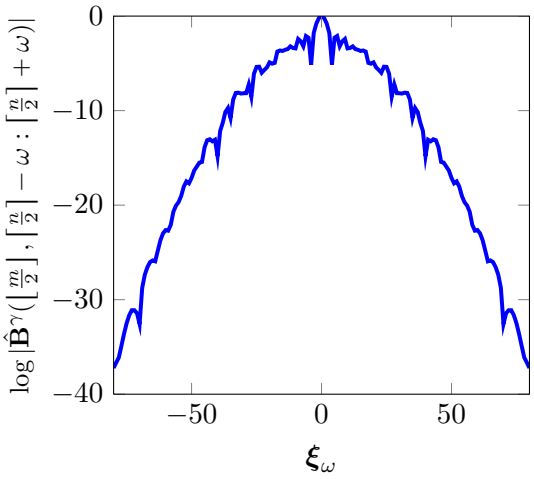
containing a section from the center of $\boldsymbol{\xi}$.



(a) Blurred Image $\mathbf{B}$



(b) Entries of $\log |\hat{\mathbf{B}}^{\gamma}(\lfloor \frac{m}{2} \rfloor, 1 : n)|$ plotted against $\boldsymbol{\xi}$. The area above noise level is located between the green lines.



(c) Entries of $\log \left| \hat{\mathbf{B}}^{\gamma} \left(\lfloor \frac{m}{2} \rfloor, \lceil \frac{n}{2} \rceil - \omega : \lceil \frac{n}{2} \rceil + \omega \right) \right|$ plotted against $\boldsymbol{\xi}_{\omega}$

Figure 3.2.: Blurred image (a), corresponding ray through the logarithmized Fourier transform of $\mathbf{B}$ (b), and the area in (b) above noise level (c).

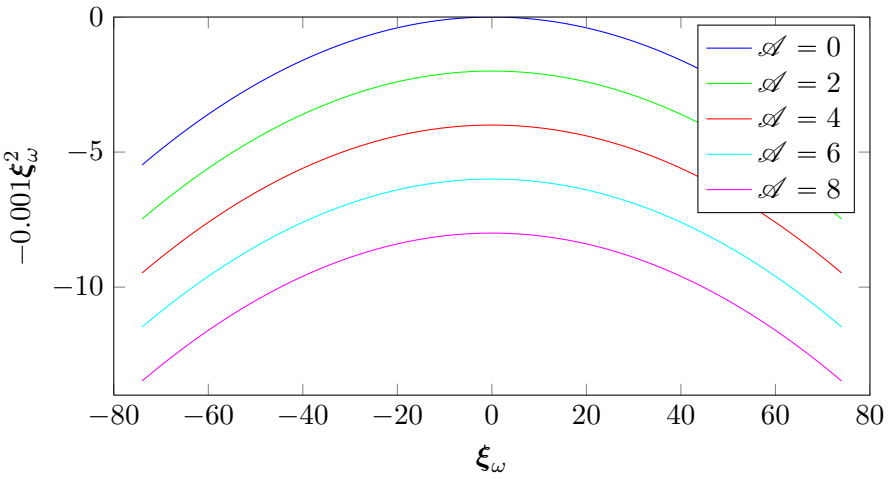


Figure 3.3.: $-\alpha(\boldsymbol{\xi}_\omega)^{2\beta} - \mathcal{A}$ plotted against $\boldsymbol{\xi}_\omega$ for different $\mathscr{A}$, where $\alpha = 0.001$ and $\beta = 1$.

That is, for the horizontal line through the center, restricted to this interval, we get

$$\begin{aligned}&\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega : \left\lceil\frac{n}{2}\right\rceil + \omega\right)\right| \\ &= \underbrace{-\alpha\boldsymbol{\Xi}^{2\beta}\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega : \left\lceil\frac{n}{2}\right\rceil + \omega\right)}_{-\alpha\boldsymbol{\xi}_\omega^{2\beta}} + \log\left|\hat{\mathbf{X}}^\gamma\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega : \left\lceil\frac{n}{2}\right\rceil + \omega\right)\right|\end{aligned} \tag{3.8}$$

where the expression for the mid-row of $\hat{\mathbf{P}}$ simplifies through substitution by vector $\boldsymbol{\xi}_\omega$. One observes that for the example image illustrated in Figure 3.2, the part of the plot in (b) that is located above noise level lies approximately in the interval $[-82, 82]$, that is $\omega \approx 82$. For the reconstruction of the PSF, we only regard this cut-out above noise level which is shown in Figure 3.2 (c).

As we do not know the true image, we have no information about $\log|\hat{\mathbf{X}}^\gamma|$ besides that the array elements lie on the interval $]-\infty, 0]$. Therefore, the last summand in (3.8) is replaced by a constant $-\mathcal{A} = -\mathscr{A}\mathbf{1}_{2\omega+1}^T \in \mathbb{R}^{1\times(2\omega+1)}$ yielding equation

$$\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega : \left\lceil\frac{n}{2}\right\rceil + \omega\right)\right| = -\alpha\boldsymbol{\xi}_\omega^{2\beta} - \mathcal{A} \tag{3.9}$$

where $\mathscr{A} > 0$ is a scalar. Plotting the right hand side of (3.9), $-\alpha\boldsymbol{\xi}_\omega^{2\beta} - \mathcal{A}$, against $\boldsymbol{\xi}_\omega$, scalar $\mathscr{A}$ defines the apex of the curve, as illustrated in Figure 3.3. The name *APEX method* is deduced from this fact. Note that introducing $\mathscr{A}$ implies that the equation is no longer valid for $\xi_i = 0$.

Now, to approximate the PSF we previously specify a value for $\mathscr{A}$ and obtain parameters α and β by solving the nonlinear least squares problem

$$\min_{\alpha,\beta}\left\|\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega : \left\lceil\frac{n}{2}\right\rceil + \omega\right)\right| + \alpha\boldsymbol{\xi}_\omega^{2\beta} + \mathcal{A}\right\|_2^2 \tag{3.10}$$

that can be substituted into expression (3.1) yielding an estimate of the PSF for the blind deblurring problem.

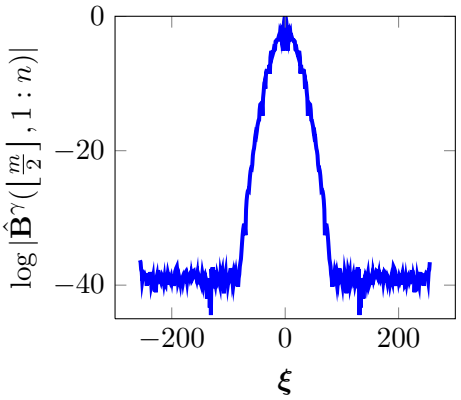
Below, we give an example for image reconstruction using a point spread function estimate computed by means of the APEX method. Moreover, we expand on this method for the case when $\beta = 1$, that is for Gaussian PSFs.

Reconstructing x The APEX method yields an approximation of the PSF matrix $\mathbf{A}$ causing the blur in the observed image $\mathbf{b}$. Hence, the blind deconvolution problem simplifies to a deconvolution problem discussed in Chapter 2 where only $\mathbf{x}$ is unknown. According to Section 2.4, we solve the least squares problem directly using Tikhonov regularization with generalized cross-validation.

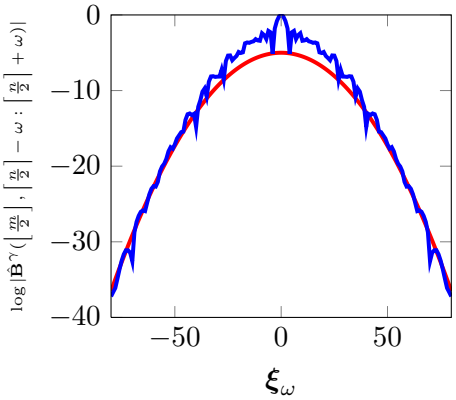
In Figure 3.4, we give an example of deblurring the image from Figure 3.2 using the APEX method. First, $\log|\hat{\mathbf{B}}^{\gamma}\left(\left\lfloor\frac{m}{2}\right\rfloor, 1:n\right)|$ is plotted against $\boldsymbol{\xi}$, as illustrated in subfigure (a). The part of the plot in (a) that is located above the noise level is displayed in subfigure (b). In subfigure (b), the cut-out plot is approximated by $-\alpha\boldsymbol{\xi}_{\omega}^{2\beta} - \mathcal{A}$ plotted against $\boldsymbol{\xi}_{\omega}$, which is equivalent to solving optimization problem (3.10). Subsequently, the obtained parameters α and β are used to calculate the PSF. Subfigure (c) illustrates horizontal intersections through both the true and reconstructed PSF. Both plots overlap almost completely. To the right of the plot of intersection lines, images of the true and reconstructed PSF are shown in subfigures (d) and (e). Having determined the PSF applying the APEX method, we receive the deblurred image solving a non-blind deconvolution problem. True, blurred and deblurred image are compared at the bottom of Figure 3.4.

Implementation We implemented the APEX method including a MATLAB GUI to interactively solve the curve fitting problem. Figure 3.5 shows the graphical user interface for deblurring problems involving rotational symmetric Gaussian PSFs. The blue curve on the left hand side depicts, as in the previous part of this section, the part of the horizontal ray through the center of $\log|\hat{\mathbf{B}}^{\gamma}|$ above noise level. On the left bottom, apex $\mathscr{A}$, that is the height of the approximated curve, can be controlled. After a value for $\mathscr{A}$ has been chosen, a nonlinear least squares fit is executed utilizing MATLAB's `lsqcurvefit` function. The resulting approximation is shown as the red curve. If the approximation is satisfactory, pressing the button "Use this Approximation" initiates the computation of the point spread function with parameter α, and finally the deblurring of the image that is displayed on the right hand side. Controlling the apex through a GUI allows a very precise adjustment which is crucial since even small changes in the apex can result in huge changes in the deblurred image, which is illustrated by a sequence of images deblurred with different $\mathscr{A}$ in Appendix A.2.

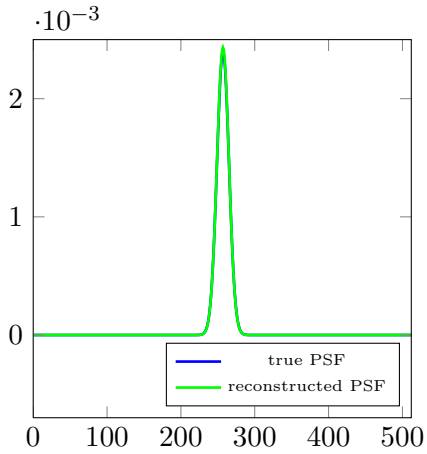
So far, the APEX method can only be used if the point spread function to be reconstructed is rotationally symmetric. In the next paragraph, we extend the APEX



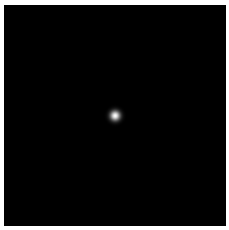
(a) Entries of $\log|\hat{\mathbf{B}}^{\gamma}(\lfloor\frac{m}{2}\rfloor, 1:n)|$ plotted against $\boldsymbol{\xi}$



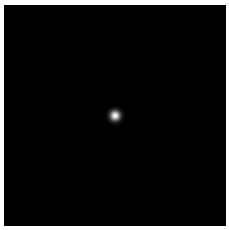
(b) Interval of $\log|\hat{\mathbf{B}}^{\gamma}(\lfloor\frac{m}{2}\rfloor, 1 : n)|$ above noise level, and approximating curve



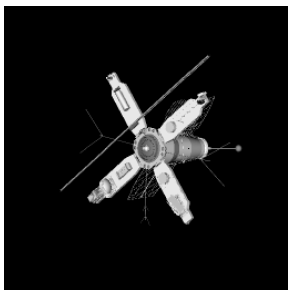
(c) PSF intersection line



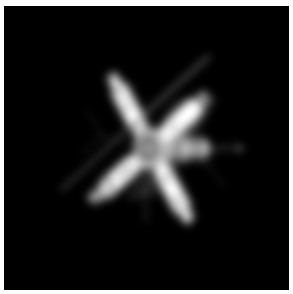
(d) True PSF



(e) Reconstructed PSF



(f) True Image



(g) Blurred Image

(h) Reconstructed Image

Figure 3.4.: Approximation of the PSF using $\log|\hat{\mathbf{B}}^{\gamma}(\lfloor\frac{m}{2}\rfloor, 1 : n)|$, and subsequent reconstruction of the true image.

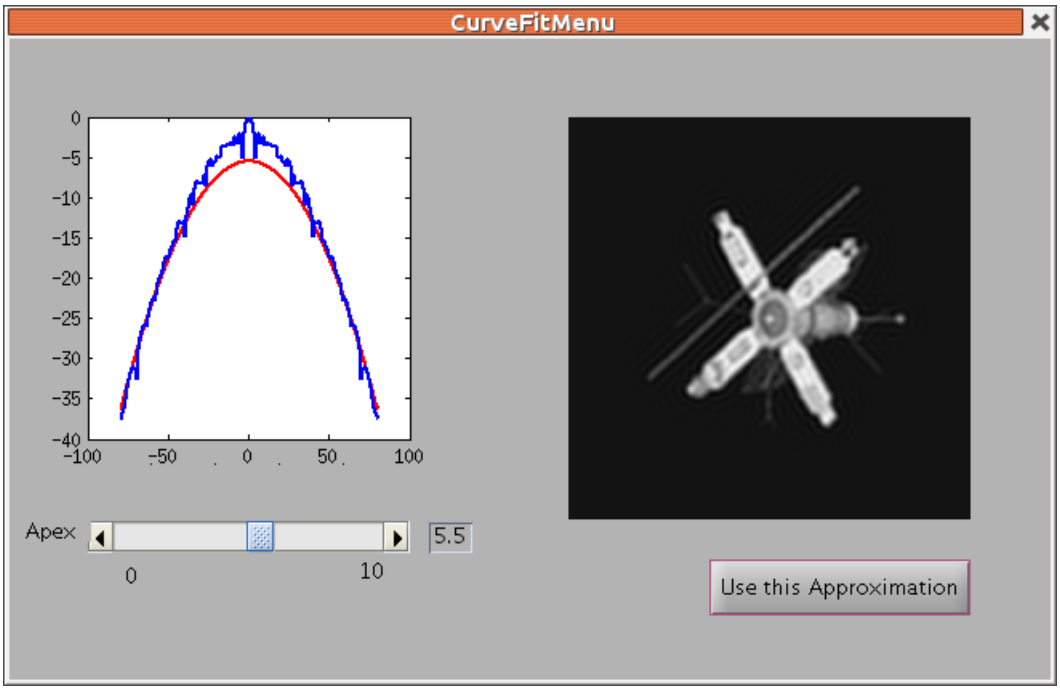


Figure 3.5.: MATLAB GUI for performing image deblurring using the APEX method.

method to double symmetric Gaussian PSFs.

Generalization for Gaussian PSFs Within the framework of this thesis we confined our studies to Gaussian point spread functions. Thus, $\beta = 1$ and function (3.1) simplifies to

$$\hat{P}_\alpha(\xi, \eta) = \exp\left\{-\alpha\left(\xi^2 + \eta^2\right)\right\} \tag{3.11}$$

for $\alpha > 0$. Since we are not only interested in rotational symmetric PSFs ($s_1 = s_2$, $\rho = 0$; see Section 2.2), we generalize expression (3.11) to the double symmetric formulation

$$\hat{P}_{\alpha_1,\alpha_2}(\xi, \eta) = \exp\left\{-\left(\alpha_1\xi^2 + \alpha_2\eta^2\right)\right\} \tag{3.12}$$

where $\alpha_1, \alpha_2 > 0$ ($s_1 \neq s_2$, $\rho = 0$; see Section 2.2).

Using the notation from above, we can rewrite $\mathbf{B} = \mathbf{P} * \mathbf{X}$ as

$$\log|\hat{\mathbf{B}}^\gamma| = -\left(\alpha_1\boldsymbol{\Xi}^2 + \alpha_2\mathbf{H}^2\right) + \log|\hat{\mathbf{X}}^\gamma|$$

according to (3.6). Compared to the Gaussian rotational symmetric case, there are not only one (α), but two (α_1, α_2) unknowns to be searched for in order to estimate the PSF. As previously, we consider the horizontal line through the center in interval $[-\omega_1, \omega_1]$ above noise level

$$\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega_1 : \left\lceil\frac{n}{2}\right\rceil + \omega_1\right)\right| \approx -\alpha_1\boldsymbol{\xi}_{\omega_1}^2 - \mathcal{A}_1$$

where $\mathcal{A}_1 = \mathscr{A}_1 \mathbf{1}_{2\omega_1+1}^T$, $\mathscr{A}_1 \in \mathbb{R}_+$, resulting in an expression that involves only α_1. We obtain an expression that only contains parameter α_2, but not α_1, by setting up an expression for the vertical line through the center,

$$\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lceil\frac{m}{2}\right\rceil - \omega_2 : \left\lceil\frac{m}{2}\right\rceil + \omega_2, \left\lfloor\frac{n}{2}\right\rfloor\right)\right| \approx -\alpha_2 \boldsymbol{\eta}_{\omega_2}^2 - \mathcal{A}_2,$$

with $\mathcal{A}_2 = \mathscr{A}_2 \mathbf{1}_{2\omega_2+1}^T$, $\mathscr{A}_2 \in \mathbb{R}_+$, and where $\boldsymbol{\eta}_{\omega_2} = (-\omega_2, \ldots, \omega_2) \in \mathbb{N}^{1\times(2\omega_2+1)}$ according to $\boldsymbol{\xi}_{\omega_1}$.

Thus, we have to solve two similar least squares problems

$$\min_{\alpha_1}\left\|\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lfloor\frac{m}{2}\right\rfloor, \left\lceil\frac{n}{2}\right\rceil - \omega_1 : \left\lceil\frac{n}{2}\right\rceil + \omega_1\right)\right| + \alpha_1 \boldsymbol{\xi}_{\omega_1}^2 + \mathcal{A}_1\right\|_2^2$$
$$\min_{\alpha_2}\left\|\log\left|\hat{\mathbf{B}}^\gamma\left(\left\lceil\frac{m}{2}\right\rceil - \omega_2 : \left\lceil\frac{m}{2}\right\rceil + \omega_2, \left\lfloor\frac{n}{2}\right\rfloor\right)\right| + \alpha_2 \boldsymbol{\eta}_{\omega_2}^2 + \mathcal{A}_2\right\|_2^2$$

that are independent of each other. The estimated values for α_1 and α_2 are then substituted in (3.12) to calculate the PSF.

We designed a GUI similar to the one in Figure 3.5 for the double symmetric case adding a second box on the left hand side where the second parameter can be chosen. A detailed description of experiments and results is given in Section 5.2.

3.2. Simultaneous Computation of PSF and Reconstructed image

In this section we present two methods to simultaneously calculate an approximation of the point spread function and sharp image $\mathbf{x}$. As an introduction to this topic, we first consider the simple and intuitive alternating least squares (ALS) method. Second, a more complicated yet more effective method, using the Gauss-Newton method, is introduced. Both approaches, the alternating least squares method and the Gauss-Newton approach, are iterative methods that require an initial value for the PSF. Furthermore, we include Tikhonov regularization (Section 2.4.2) in both algorithms due to ill-posedness of the deconvolution problem and impose periodic boundary conditions. A flowchart in Figure 3.6 illustrates the general approach to iteratively solve blind deconvolution problems.

3.2.1. Alternating Least Squares Method

The alternating least squares method is a simple algorithm to solve (2.21) where approximations for the true image and PSF are computed in turn. First, we specify an

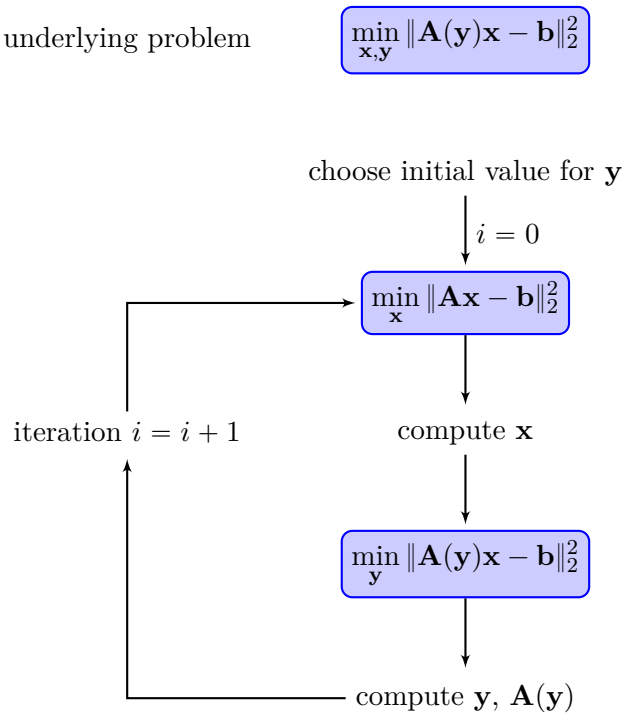


Figure 3.6.: Iterative model for solving the blind deconvolution problem. $\mathbf{A}$ and $\mathbf{x}$ are computed simultaneously.

initial value for parameters that define a PSF, for instance s_1, s_2 and ρ for a Gaussian PSF. Then, Tikhonov regularization with generalized cross-validation is applied to solve the linear least squares problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \min_{\hat{\mathbf{x}}} \|\mathbf{\Lambda}\hat{\mathbf{x}} - \hat{\mathbf{b}}\|_2^2.$$

We can calculate $\hat{\mathbf{x}}$ very efficiently from the Fourier representation of the problem and subsequently use the inverse Fourier transform to obtain $\mathbf{x}$. Next, we again apply Tikhonov regularization with generalized cross-validation but to solve a slightly different linear least squares problem

$$\min_{\mathbf{A}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \min_{\mathbf{\Lambda}} \|\mathbf{\Lambda}\hat{\mathbf{x}} - \hat{\mathbf{b}}\|_2^2$$

which is equivalent to

$$\min_{\boldsymbol{\lambda}} \|\mathrm{diag}(\hat{\mathbf{x}})\boldsymbol{\lambda} - \hat{\mathbf{b}}\|_2^2$$

where $\boldsymbol{\lambda} = \mathrm{diag}(\mathbf{\Lambda}) = (\lambda_1, \ldots, \lambda_N)^T$. That means, the nonlinear least squares problem $\min_{\mathbf{y}} \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2$ is regarded as the linear least squares problem $\min_{\boldsymbol{\lambda}} \|\mathrm{diag}(\hat{\mathbf{x}})\boldsymbol{\lambda} - \hat{\mathbf{b}}\|_2^2$. Both least squares approximation steps are repeated until a satisfying solution is achieved. The ALS method is illustrated in pseudocode in Algorithm 3.1.

Alternating Least Squares

```
function x = ALS(A,b)

choose initial guess A to the PSF
choose number of iterations n
for i = 1,...,n
    ε1 = GCV(A,b)                    % see Algorithm 2.1 ...
    x̂ = Tikhonov(Λ,b̂,ε1)             % for GCV and Tikhonov
    x = F*x̂
    ε2 = GCV(diag(x),b)
    λ = Tikhonov(diag(x̂),b̂,ε2)
    A = F*ΛF
end

```

Algorithm 3.1

3.2.2. Gauss-Newton Method

Another approach for blind deconvolution uses the Gauss-Newton method to solve the nonlinear subproblem $\min_{\mathbf{y}} \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2$. We give a general introduction into the Gauss-Newton method, and subsequently explain the overall blind deconvolution algorithm.

The Gauss-Newton method has proved to be worthwhile in solving an optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad f : \mathbb{R}^n \to \mathbb{R} \tag{3.13}$$

for an objective f having a special structure

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{r}(\mathbf{x})\|_2^2$$

with $\mathbf{x} \in \mathbb{R}^n$ and residual $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$ [25, p.254ff.].

We can motivate the Gauss-Newton method as follows: The residual is approximated by a Taylor expansion of first order at a fixed position $\mathbf{x}_k$

$$\begin{aligned} f(\mathbf{x}_k + \mathbf{d}_k) &= \frac{1}{2}\|\mathbf{r}(\underbrace{\mathbf{x}_k + \mathbf{d}_k}_{\mathbf{x}_{k+1}})\|_2^2 \\ &\approx \frac{1}{2}\|\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_\mathbf{r}(\mathbf{x}_k)\mathbf{d}_k\|_2^2 =: m_k(\mathbf{d}_k) \end{aligned} \tag{3.14}$$

where $\mathbf{J_r}$ describes the Jacobian matrix

$$\mathbf{J_r}(\mathbf{x}) = \begin{pmatrix} \frac{\partial r_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial r_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial r_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

of residual $\mathbf{r}(\mathbf{x})$. We search for a vector $\mathbf{d}_k$ minimizing (3.14), that is we want to solve the optimization problem

$$\min_{\mathbf{d}_k} m_k(\mathbf{d}_k).$$

Therefore, the gradient of m_k with respect to $\mathbf{d}_k$,

$$\nabla m_k(\mathbf{d}_k) = \mathbf{J}_\mathbf{r}^T(\mathbf{x}_k)\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_r^T(\mathbf{x}_k)\mathbf{J_r}(\mathbf{x}_k)\mathbf{d}_k,$$

is set equal to zero. Rearranging yields a system of equations

$$\mathbf{J}_r^T(\mathbf{x}_k)\mathbf{J_r}(\mathbf{x}_k)\mathbf{d}_k = -\mathbf{J}_\mathbf{r}^T(\mathbf{x}_k)\mathbf{r}(\mathbf{x}_k). \tag{3.15}$$

The vector $\mathbf{d}_k \in \mathbb{R}^n$, obtained through solving (3.15), minimizes m_k, but not exactly $f(\mathbf{x}_k + \mathbf{d}_k)$ due to approximations in (3.14). However, we suppose that $\mathbf{d}_k$ indicates the direction leading to a minimum and choose it as the so called *descent direction*. Starting at $\mathbf{x}_k$ we search along direction $\mathbf{d}_k$ for a vector $\mathbf{x}_{k+1}$ that yields a smaller value for $f(\mathbf{x}_k + \mathbf{d}_k)$, that is we determine a scalar $a_k \in \mathbb{R}, a_k > 0$, and define

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k\mathbf{d}_k$$

such that

$$f(\mathbf{x}_{k+1}) \leqslant f(\mathbf{x}_k).$$

Scalar a_k indicates how far we have to walk along descent direction $\mathbf{d}_k$ to decrease the function value as much as possible. a_k is called *step length* and can be computed using *line search* methods described in the book by Nocedal and Wright [25, p.30ff.].

Considering the blind deconvolution problem supposing that what causes the blur is a Gaussian PSF with parameters $\mathbf{y} = (s_1, s_2, \rho)^T$ the optimization problem according to (3.13) is given by

$$\min_{\mathbf{y}} \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2$$

and the residual is defined as $\mathbf{r}(\mathbf{y}) = \mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}_+^{N\times N}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}_+^{N\times 1}$. Thus, in this particular case the Jacobian matrix is set up as

$$\mathbf{J_r}(\mathbf{y}) = \begin{pmatrix} \frac{\partial r_1(\mathbf{y})}{\partial s_1} & \frac{\partial r_1(\mathbf{y})}{\partial s_2} & \frac{\partial r_1(\mathbf{y})}{\partial \rho} \\ & \vdots & \\ \frac{\partial r_N(\mathbf{y})}{\partial s_1} & \frac{\partial r_N(\mathbf{y})}{\partial s_2} & \frac{\partial r_N(\mathbf{y})}{\partial \rho} \end{pmatrix}$$

where $\mathbf{r} \in \mathbb{R}^N$. We use *finite differences* to approximate the partial derivatives (see [15, p.629 ff.]).

Algorithm 3.2 describes solving the blind deconvolution problem using the Gauss-Newton method in pseudocode. Lines 3 to 4 describe the initialization where an initial guess for $\mathbf{y}$ is chosen and $\mathbf{A}(\mathbf{y})\mathbf{x} = \mathbf{b}$ is solved for $\mathbf{x}$ using Tikhonov regularization with GCV. Then, for a preset number of iterations (line 5 to 15), the following is executed: The descent direction $\mathbf{d}$ is determined solving $\mathbf{J_r}^T\mathbf{J_r}\mathbf{d} = -\mathbf{J_r}^T\mathbf{r}$. We choose the step length to be equal to one and update $\mathbf{y}$ in line 8. Subsequently, $\mathbf{x}$ is computed again applying Tikhonov regularization with GCV. A simple line search is performed in lines 10 to 14 where in every iteration of the while-loop the step length is divided by two, $\mathbf{y}$ is updated and a new vector $\mathbf{x}$ is determined.

In Section 5.1, we give some results for applying the Gauss-Newton method to deconvolution problems.

Gauss-Newton Method

```
1   function [x,A(y)] = GaussNewton(A(y),b)
2
3   choose initial guess for y = y_0
4   choose number of iterations n
5   r = computeResidual(y_0,b)
6   for i = 1,...,n
7      solve J_r^T(y_{i-1})J_r(y_{i-1})d = -J_r^T(y_{i-1})r for d  % descent direction
8      a = 1                                                      % step length
9      y_i = y_{i-1} + ad
10     r = computeResidual(y_i,b)
11     while f(y_i) > f(y_{i-1})                                  % line search
12         a = a/2
13         y_i = y_{i-1} + ad
14         r = computeResidual(y_i,b)
15     end
16  end
17
18
19  function r = computeResidual(y,b)
20
21  ε = GCV(A(y),b)                          % see Algorithm 2.1 ...
22  x̂ = Tikhonov(Λ(y),b̂,ε)                   % for Tikhonov and GCV
23  x = F*x̂
24  r = b - A(y)x
```

Algorithm 3.2

4. Multiframe Blind Deconvolution

A blind deconvolution problem where multiple blurred images $\mathbf{b}_j \in \mathbb{R}_+^{mn\times 1}$, $j = 1, \ldots, p$ of the same object have to be taken into account is called multiframe blind deconvolution. Combining Sections 2.5 and 2.6, multiframe blind deconvolution seeks to find a solution for the extremely underdetermined optimization problem

$$\min_{\mathbf{x},\mathbf{y}_j} \sum_{j=1}^{p} \|\mathbf{A}(\mathbf{y}_j)\mathbf{x} - \mathbf{b}_j\|_2^2 = \min_{\mathbf{x},\mathbf{y}_j} \left\| \begin{pmatrix} \mathbf{A}(\mathbf{y}_1) \\ \vdots \\ \mathbf{A}(\mathbf{y}_p) \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_p \end{pmatrix} \right\|_2^2, \tag{4.1}$$

in which not only the true image $\mathbf{x} \in \mathbb{R}_+^{mn\times 1}$, but also all p point spread function matrices $\mathbf{A}_j := \mathbf{A}(\mathbf{y}_j) \in \mathbb{R}_+^{mn\times mn}$, $j = 1, \ldots, p$ are unknown.

One way of solving the multiframe blind deconvolution problem is to use the Gauss-Newton method with Tikhonov regularization and GCV. The algorithm for multiframe blind deconvolution differs slightly from Algorithm 3.2 that deals with the singleframe blind deconvolution problem. For instance, let us assume that the blur is induced by Gaussian PSFs such that $\mathbf{y}_j = (s_1^{(j)}, s_2^{(j)}, \rho^{(j)})^T$ for $j = 1, \ldots, p$. Then, vector $\mathbf{y} = (\mathbf{y}_1; \ldots; \mathbf{y}_p) \in \mathbb{R}^{3p\times 1}$ contains the $\mathbf{y}_j$ one below the other. The Gauss-Newton algorithm requires an initial value $\mathbf{y}_0$ for $\mathbf{y}$. Given $\mathbf{y}$, the approximated true image is obtained solving problem (4.1) for $\mathbf{x}$ using Tikhonov regularization with GCV. Moreover, $\mathbf{y}$ is updated as stated in the algorithm in Section 3.2.2 where the Jacobian matrix now has the size $mnp \times 3p$ and is given by

$$\mathbf{J}_\mathbf{r}(\mathbf{y}) = \left(\frac{\partial r_s(\mathbf{y})}{\partial \mathbf{y}(t)} \right)_{\substack{s=1,\ldots,mnp \\ t=1,\ldots,3p}}$$

with residual $\mathbf{r} \in \mathbb{R}^{mnp\times 1}$.

The ALS method from Section 3.2.1 can be expanded to be used in the multiframe case as well. However, we cannot recommend to apply the APEX method introduced in Section 3.1.1 to the multiframe blind deconvolution problem. Using APEX, every PSF has to be computed individually yielding a multiframe deconvolution problem as given in (2.18). Especially for a large number of frames, approximating the point spread functions is a cumbersome procedure since the APEX parameter has to be adjusted independently for every frame. Besides, another restriction is that we can only use the APEX method for Gaussian point spread functions where $\rho = 0$.

In the following, we want to take a short look at the structure of the linear system of equations that is solved using Tikhonov regularization to update $\mathbf{x}$ during each Gauss-Newton iteration. Similar to Section 2.3.2 the system of equations can be stated in the Fourier domain

$$\begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix} \hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{b}}_1 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix}, \tag{4.2}$$

assuming periodic boundary conditions. The Tikhonov method solves the normal equation of (4.2) for $\hat{\mathbf{x}}$. The corresponding system of equations

$$\begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix}^* \begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix} \hat{\mathbf{x}} = \begin{pmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_p \end{pmatrix}^* \begin{pmatrix} \hat{\mathbf{b}}_1 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix}$$

is underdetermined consisting of $(p+1)mn$ unknowns and mn equations. In the following we describe methods to simplify expression (4.1), provided the point spread functions in our model satisfy certain characteristics.

4.1. Compact Multiframe Blind Deconvolution

Hope and Jefferies [18] introduce a method to reduce inverse problem (4.1) to a problem where only one of the $\mathbf{A}(\mathbf{y}_j)$, $j = 1, \ldots, p$ and $\mathbf{x}$ have to be determined. Therefore, they assume that there exists one PSF whose Fourier transform is nonzero everywhere that the Fourier transform of $\mathbf{x}$ is nonzero. As a result of the decreased underdeterminedness of the multiframe blind deconvolution problem, computational costs can be reduced which is especially helpful for large data sets.

Assuming periodic boundary conditions as defined in Section 2.3.2 allows us to express $\mathbf{A}(\mathbf{y}_j)$ by its spectral decomposition $\mathbf{A}(\mathbf{y}_j) = \mathbf{F}^* \mathbf{\Lambda}(\mathbf{y}_j)\mathbf{F}$. In the following, let $\mathbf{A}_j := \mathbf{A}(\mathbf{y}_j)$ and $\mathbf{\Lambda}_j := \mathbf{\Lambda}(\mathbf{y}_j)$ for simplicity. Comparing matrices $\mathbf{\Lambda}_j$, we denote the matrix which is nonzero at every position where the other $\mathbf{\Lambda}_j$ have nonzero entries with $\mathbf{\Lambda}_k$, supposing that such a matrix exists. Then, frame k is referred to as the *control frame*. Mathematically speaking,

$$(\mathbf{\Lambda}_k)_{i,i} \neq 0 \text{ if there exists a } j \text{ such that } (\mathbf{\Lambda}_j)_{i,i} \neq 0$$

where $(\mathbf{\Lambda}_k)_{i,i}$ and $(\mathbf{\Lambda}_j)_{i,i}$ describe the i-th diagonal element, respectively.

Then

$$\begin{aligned} \mathbf{A}_j &= \mathbf{F}^* \mathbf{\Lambda}_j \mathbf{F} \\ &= \mathbf{F}^* \mathbf{\Lambda}_j \mathbf{\Lambda}_k^+ \mathbf{\Lambda}_k \mathbf{F} \end{aligned}$$

where

$$\mathbf{\Lambda}_k^+ = \begin{cases} \frac{1}{\lambda_i^{(k)}} & \text{if } \lambda_i^{(k)} \neq 0 \\ 0 & \text{if } \lambda_i^{(k)} = 0 \end{cases}$$

is the pseudoinverse of $\mathbf{\Lambda}_k$. Hence, we can write the objective of optimization problem (4.1) as

$$\begin{aligned} \sum_{j=1}^{p} \|\mathbf{A}_j\mathbf{x} - \mathbf{b}_j\|_2^2 &= \|\mathbf{A}_k\mathbf{x} - \mathbf{b}_k\|_2^2 + \sum_{\substack{j=1 \\ j\neq k}}^{p} \|\mathbf{A}_j\mathbf{x} - \mathbf{b}_j\|_2^2 \\ &= \|\mathbf{F}^*\mathbf{\Lambda}_k\mathbf{F}\mathbf{x} - \mathbf{b}_k\|_2^2 + \sum_{\substack{j=1 \\ j\neq k}}^{p} \|\mathbf{F}^*\mathbf{\Lambda}_j\mathbf{\Lambda}_k^+\mathbf{\Lambda}_k\mathbf{F}\mathbf{x} - \mathbf{b}_j\|_2^2 \\ &= \|\mathbf{\Lambda}_k\hat{\mathbf{x}} - \hat{\mathbf{b}}_k\|_2^2 + \sum_{\substack{j=1 \\ j\neq k}}^{p} \|\mathbf{\Lambda}_j\mathbf{\Lambda}_k^+\mathbf{\Lambda}_k\hat{\mathbf{x}} - \hat{\mathbf{b}}_j\|_2^2. \end{aligned} \tag{4.3}$$

From Section 2.3.2 we know that the Fourier representation of the blurring process is given by $\hat{\mathbf{b}}_j = \mathbf{\Lambda}_j\hat{\mathbf{x}}$ and define $\text{diag}(\hat{\mathbf{b}}_j) = \text{diag}\left(\left(\hat{b}_1^{(j)}, \ldots, \hat{b}_{mn}^{(j)}\right)^T\right)$ as the diagonal matrix containing vector $\hat{\mathbf{b}}_j$ and

$$\text{diag}(\hat{\mathbf{b}}_j)^+ = \begin{cases} \frac{1}{\hat{b}_i^{(j)}} & \text{if } \hat{b}_i^{(j)} \neq 0 \\ 0 & \text{if } \hat{b}_i^{(j)} = 0 \end{cases}$$

as its pseudoinverse, where $i = 1, \ldots, mn$. Similarly, let $\text{diag}(\hat{\mathbf{x}}) = \text{diag}\left((\hat{x}_1, \ldots, \hat{x}_{mn})^T\right)$ and

$$\text{diag}(\hat{\mathbf{x}})^+ = \begin{cases} \frac{1}{\hat{x}_i} & \text{if } \hat{x}_i \neq 0 \\ 0 & \text{if } \hat{x}_i = 0 \end{cases}$$

hold for vector $\hat{\mathbf{x}}$. Then we can state the Fourier representation of the blurring process as

$$\begin{aligned} \mathbf{\Lambda}_j\text{diag}(\hat{\mathbf{x}}) &= \text{diag}(\hat{\mathbf{b}}_j) \\ \Rightarrow \mathbf{\Lambda}_j &= \text{diag}(\hat{\mathbf{b}}_j)\text{diag}(\hat{\mathbf{x}})^+ \end{aligned} \tag{4.4}$$

and in the same way

$$\begin{aligned} \mathbf{\Lambda}_k\text{diag}(\hat{\mathbf{x}}) &= \text{diag}(\hat{\mathbf{b}}_k) \\ \Rightarrow \mathbf{\Lambda}_k^+ &= \text{diag}(\hat{\mathbf{b}}_k)^+\text{diag}(\hat{\mathbf{x}}) \end{aligned} \tag{4.5}$$

for $j = k$. Therefore, the product of (4.4) and (4.5) can be written as

$$\mathbf{\Lambda}_j\mathbf{\Lambda}_k^+ = \text{diag}(\hat{\mathbf{b}}_j)\text{diag}(\hat{\mathbf{b}}_k)^+$$

since the nonzero entries of $\text{diag}(\hat{\mathbf{x}})$ divide out.

Substituting into (4.3) yields

$$\|\mathbf{\Lambda}_k\hat{\mathbf{x}} - \hat{\mathbf{b}}_k\|_2^2 + \sum_{\substack{j=1 \\ j\neq k}}^{p} \|\mathrm{diag}(\hat{\mathbf{b}}_j)\mathrm{diag}(\hat{\mathbf{b}}_k)^+\mathbf{\Lambda}_k\hat{\mathbf{x}} - \hat{\mathbf{b}}_j\|_2^2. \tag{4.6}$$

where only $\hat{\mathbf{x}}$ and $\mathbf{\Lambda}_k$ are unknown. We rearrange (4.6) to obtain an expression consisting of only one norm, similar to the multiframe problem in (4.1). For simplicity of writing we suppose that $k = 1$. Then, in the Fourier domain, the least squares problem with objective (4.6) can be stated as

$$\min_{\hat{\mathbf{x}},\mathbf{y}_k} \left(\left\| \begin{pmatrix} \mathbf{I} \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_p \end{pmatrix} \mathbf{\Lambda}(\mathbf{y}_k)\hat{\mathbf{x}} - \begin{pmatrix} \hat{\mathbf{b}}_k \\ \hat{\mathbf{b}}_2 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix} \right\|_2^2 \right),$$

where $\mathbf{D}_j, j = 2, \ldots, p$ is defined by the product of diagonal matrices

$$\mathbf{D}_j = \mathrm{diag}(\hat{\mathbf{b}}_j)\mathrm{diag}(\hat{\mathbf{b}}_k)^+ \text{ for } j \neq k.$$

Again, one possibility of solving for true image $\mathbf{x}$ and point spread function of control frame k is to apply the Gauss-Newton method with Tikhonov regularization. The objective of the compact problem consists of merely one PSF implying that $\mathbf{y} \in \mathbb{R}^{3\times 1}$ regarding Gaussian blur. Residual $\mathbf{r}$, however, is still as large as for the former multiframe problem resulting in a Jacobian matrix

$$\mathbf{J}_\mathbf{r}(\mathbf{y}) = \left(\frac{\partial r_s(\mathbf{y})}{\partial \mathbf{y}(t)} \right)_{\substack{s=1,\ldots,mnp \\ t=1,\ldots,3}}$$

of size $mnp \times 3$. We can also use ALS to approach the compact multiframe blind deconvolution problem, and APEX is suitable as well, since only one point spread function has to be estimated.

The normal equation used to solve the problem for $\mathbf{x}$ is given by

$$\left(\begin{pmatrix} \mathbf{I} \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_p \end{pmatrix} \mathbf{\Lambda}_k \right)^* \begin{pmatrix} \mathbf{I} \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_p \end{pmatrix} \mathbf{\Lambda}_k\hat{\mathbf{x}} = \left(\begin{pmatrix} \mathbf{I} \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_p \end{pmatrix} \mathbf{\Lambda}_k \right)^* \begin{pmatrix} \hat{\mathbf{b}}_k \\ \hat{\mathbf{b}}_2 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix}. \tag{4.7}$$

That is, the system of equations has $2mn$ unknowns and mn equations. In the next section we illustrate that the problem can be reduced even further, into a system with only a single frame.

4.2. Reduction to Single Frame Blind Deconvolution

Above, assuming the existence of a control frame the unknowns in the multiframe blind deconvolution problem could be reduced from p point spread functions and true image $\mathbf{x} \in \mathbb{R}_+^{mn\times 1}$ to only a single point spread function and $\mathbf{x}$. The large matrix in expression (4.7), $(\mathbf{I}; \mathbf{D}_2; \ldots; \mathbf{D}_p) \in \mathbb{C}^{mnp\times mn}$, contains an identity matrix $\mathbf{I} \in \mathbb{R}^{mn\times mn}$ right on top and diagonal matrices $\mathbf{D}_j \in \mathbb{C}^{mn\times mn}$ arranged one below the other.

To reduce problem (4.7) further we eliminate the diagonal matrices $\mathbf{D}_2, \ldots, \mathbf{D}_p$ in this large matrix composed of diagonal matrices using *Givens rotations*.

4.2.1. Givens Rotation

In general, a Givens rotation is an orthogonal matrix

$$\mathbf{G} = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & c & \ldots & s & & & i \\ & & \vdots & \ddots & \vdots & & & \\ & & -s & \ldots & c & & & k \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & i & & k & & & \end{pmatrix} \in \mathbb{R}^{N\times N}$$

where $s = \sin(\theta)$, $c = \cos(\theta)$ for $\theta \in [0, 2\pi)$. Multiplying $\mathbf{G}^T$ with a vector $\mathbf{z} \in \mathbb{R}^N$ such that $\mathbf{G}^T\mathbf{z} = \mathbf{w}$ yields vector $\mathbf{w} \in \mathbb{R}^N$ with

$$w_j = \begin{cases} cz_i - sz_k, & \text{if } j = i, \\ sz_i + cz_k, & \text{if } j = k, \text{ and} \\ z_j, & \text{else,} \end{cases}$$

that is only two entries in $\mathbf{z}$ are modified. To induce a zero at position k in $\mathbf{w}$ we set

$$s = -\frac{z_k}{\sqrt{z_i^2 + z_k^2}} \quad \text{and}$$
$$c = \frac{z_i}{\sqrt{z_i^2 + z_k^2}}$$

resulting in $w_i = \sqrt{z_i^2 + z_k^2}$ and $w_k = 0$ [12; 32].

Generally, to force $n < N$ entries in $\mathbf{z}$ to be zero the vector is multiplied by product $\mathbf{Q}^T = \mathbf{G}_n^T \ldots \mathbf{G}_1^T$ of the particular Givens matrices such that $\mathbf{Q}^T\mathbf{z} = \mathbf{w}$. Matrix $\mathbf{Q}$ is

orthogonal as well, since it is the product of n orthogonal matrices. As with vectors, entries can be eliminated in matrices.

As mentioned above, our goal is to transform a matrix of size $mnp \times mn$ consisting of p diagonal matrices into a matrix of the same size that only contains one diagonal matrix and zeros elsewhere. Applying Givens rotations this matrix can likewise be regarded as a vector of size $mnp \times 1$, since every row contains only one entry unequal to zero. In the following, we give a small example illustrating this problem.

Consider a matrix

$$\mathbf{A} = \left(\begin{array}{cc} a_1 & 0 \\ 0 & a_2 \\ \hline a_3 & 0 \\ 0 & a_4 \end{array}\right) \in \mathbb{R}^{4\times 2}$$

comprising two diagonal matrices of size 2×2. In the following, we regard this matrix as a vector

$$\mathbf{a} = \left(\begin{array}{c} a_1 \\ a_2 \\ \hline a_3 \\ a_4 \end{array}\right) \in \mathbb{R}^{4\times 1}.$$

Then to eliminate a_3 we compute $\mathbf{a'} = \mathbf{G}_1^T\mathbf{a} = (\sqrt{a_1^2 + a_3^2}, a_2, 0, a_4)^T$ with Givens rotation matrix

$$\mathbf{G}_1^T = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ 0 & 1 & 0 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $s_1 = -\frac{a_3}{\sqrt{a_1^2+a_3^2}}$ and $c_1 = \frac{a_1}{\sqrt{a_1^2+a_3^2}}$. Subsequently, a_4 is eliminated performing rotation $\mathbf{G}_2^T\mathbf{a'} = (\sqrt{a_1^2 + a_3^2}, \sqrt{a_2^2 + a_4^2}, 0, 0)^T$ with

$$\mathbf{G}_2^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & -s_2 \\ 0 & 0 & 1 & 0 \\ 0 & s_2 & 0 & c_2 \end{pmatrix}$$

where $s_2 = -\frac{a_4}{\sqrt{a_2^2+a_4^2}}$ and $c_2 = \frac{a_2}{\sqrt{a_2^2+a_4^2}}$. The overall rotation can be described by

$$\mathbf{Q}^T\mathbf{a} = \mathbf{G}_2^T\mathbf{G}_1^T\mathbf{a} = \begin{pmatrix} \sqrt{a_1^2 + a_3^2} \\ \sqrt{a_2^2 + a_4^2} \\ 0 \\ 0 \end{pmatrix}.$$

Above, we illustrated Givens rotations as a useful tool to induce zeros in a matrix containing real values. However, applying Givens rotations to vectors or matrices with complex entries, as they might occur in the optimization problem (4.7), slightly differs from the real case. We summarize the differences in the following paragraph.

Complex-Valued Givens Rotations Eliminating entries in a complex-valued vector $\mathbf{z} \in \mathbb{C}^N$ requires a few adjustments of the Givens rotation method stated above. Complex Givens rotations are defined by unitary matrices of the form

$$\mathbf{G} = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & c & \dots & s & & & i \\ & & \vdots & \ddots & \vdots & & & \\ & & -\bar{s} & \dots & c & & & k \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & i & & k & & & \end{pmatrix} \in \mathbb{C}^{N \times N}$$

similar to the real case but where $\bar{s}$ describes the complex conjugate. The entry z_k in $\mathbf{z}$ is eliminated through multiplication by the adjoint matrix of $\mathbf{G}$ such that $\mathbf{G}^*\mathbf{z} = \mathbf{w}$. Therefore, we set

$$s = \operatorname{sign}(z_i) \frac{\bar{z}_k}{\sqrt{|z_i|^2 + |z_k|^2}} \quad \text{and}$$
$$c = \frac{|z_i|}{\sqrt{|z_i|^2 + |z_k|^2}}$$

according to Bindel et al. [4] where

$$\operatorname{sign}(z) = \begin{cases} \frac{z}{|z|} & \text{if } z \neq 0 \\ 1 & \text{else} \end{cases}$$

for $z \in \mathbb{C}$. Hence, entry $w_i = \operatorname{sign}(z_i)\sqrt{|z_i|^2 + |z_k|^2}$ and, as designated, $w_k = 0$.

4.2.2. The Compact Single Frame Blind Deconvolution Problem

As described above, Givens rotations are equivalent to finding a unitary matrix $\mathbf{Q}$ such that

$$\mathbf{Q}^* \begin{pmatrix} \mathbf{I} \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_p \end{pmatrix} = \begin{pmatrix} \mathbf{D} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix},$$

where $\mathbf{D} \in \mathbb{C}^{mn \times mn}$ is a diagonal matrix and the entries below $\mathbf{D}$ are equal to zero. Now, the unitary matrix $\mathbf{Q}^*$ is multiplied to both sides of the system of equations

$$\mathbf{Q}^* \begin{pmatrix} \mathbf{I} \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_p \end{pmatrix} \mathbf{\Lambda}_k \hat{\mathbf{x}} = \mathbf{Q}^* \begin{pmatrix} \hat{\mathbf{b}}_k \\ \hat{\mathbf{b}}_2 \\ \vdots \\ \hat{\mathbf{b}}_p \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{D} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \mathbf{\Lambda}_k \hat{\mathbf{x}} = \begin{pmatrix} \hat{\hat{\mathbf{b}}}_k \\ \hat{\hat{\mathbf{b}}}_2 \\ \vdots \\ \hat{\hat{\mathbf{b}}}_p \end{pmatrix}$$

denoting with $\hat{\hat{\mathbf{b}}}_j$ the vector $\hat{\mathbf{b}}_j$ modified by multiplication with unitary matrix $\mathbf{Q}^*$. Equivalently, the least squares problem becomes

$$\min_{\hat{\mathbf{x}}, \mathbf{y}_k} \left\| \begin{pmatrix} \mathbf{D} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \mathbf{\Lambda}(\mathbf{y}_k) \hat{\mathbf{x}} - \begin{pmatrix} \hat{\hat{\mathbf{b}}}_k \\ \hat{\hat{\mathbf{b}}}_2 \\ \vdots \\ \hat{\hat{\mathbf{b}}}_p \end{pmatrix} \right\|_2^2 = \min_{\hat{\mathbf{x}}, \mathbf{y}_k} \| \mathbf{D} \mathbf{\Lambda}(\mathbf{y}_k) \hat{\mathbf{x}} - \hat{\hat{\mathbf{b}}}_k \|_2^2, \tag{4.8}$$

that is we reduced the multiframe blind deconvolution problem to a single frame problem. As for multiframe and compact multiframe blind deconvolution we solve this problem using Gauss-Newton and the Tikhonov regularization method as illustrated in Algorithm 3.2. Alternatively, ALS or the APEX method can be applied. Similar to the compact multiframe problem from Section 4.1, the APEX method is acceptable here since only one PSF has to be approximated. In the following chapter, we illustrate some results comparing approaches to solve the three problems, multiframe blind deconvolution, compact multiframe deconvolution and compact single frame blind deconvolution.

5. Experiments and Results

5.1. Iterative Approach: Gauss-Newton Method

We solve multiframe blind deconvolution (4.3), compact multiframe blind deconvolution (4.6) and single frame blind deconvolution problem (4.8) for artificially generated data applying the Gauss-Newton method with Tikhonov regularization to the normal equation. All restorations are performed using MATLAB. To create the test data consisting of p frames, we blur one sharp image of size 256×256 with p Gaussian PSFs, and add a certain percentage of Gaussian noise to the blurred image.

We choose parameters s_1, s_2 and ρ determining width and orientation of the PSF such that $s_1^{(1)} < s_1^{(2)} < \ldots < s_1^{(p)}$, $s_2^{(1)} < s_2^{(2)} < \ldots < s_2^{(p)}$ and $\rho^{(1)} < \rho^{(2)} < \ldots < \rho^{(p)}$ implying that the PSF of the first frame is the narrowest one, as explained in Section 2.3.3, and the corresponding PSF matrix $\mathbf{A}$ has the componentwise maximum eigenvalues and hence the most nonzero (or the least very small) eigenvalues compared to the other frame's PSFs. Hence, a control frame, namely the first one, exists for this model according to Section 4.1. A point spread function data set comprising of 10 frames is shown in Figure 5.1. One observes that the s_1, s_2 and ρ specified below each image are increasing from frame to frame, and so does the width of the point spread function from PSF 1 to PSF 10.

The Gauss-Newton method requires an initial choice for the point spread functions regarding the multiframe blind deconvolution problem, and for the point spread function of the control frame if we solve the compact multiframe and the compact single frame problem. Based on experience, we set the initial guess for all point spread functions to a PSF defined by $s_1^{(0)}$, $s_2^{(0)}$ and $\rho^{(0)}$ that is slightly wider than the widest in the model stated above, such that $s_1^{(p)} < s_1^{(0)}$, $s_2^{(p)} < s_2^{(0)}$ and $\rho^{(p)} < \rho^{(0)}$.

Figure 5.2 compares the three approaches, multiframe blind deconvolution (MFBD), compact multiframe blind deconvolution (CMF) and compact single frame deconvolution (CSF), with respect to the reconstruction of true image $\mathbf{X}$. Here, we apply the Gauss-Newton algorithm with Tikhonov regularization to a data set consisting of $p = 10$ frames where no noise is added to the blurred images. The first column of Figure 5.2 shows the mean of the blurred images $\mathbf{B}_1, \ldots, \mathbf{B}_{10}$ for MFBD and CMF, and the Givens-rotated blurred image of the control frame $\tilde{\mathbf{B}}_1$. In the mid-column, the deblurred images are illustrated. Numbers above blurred and deblurred images

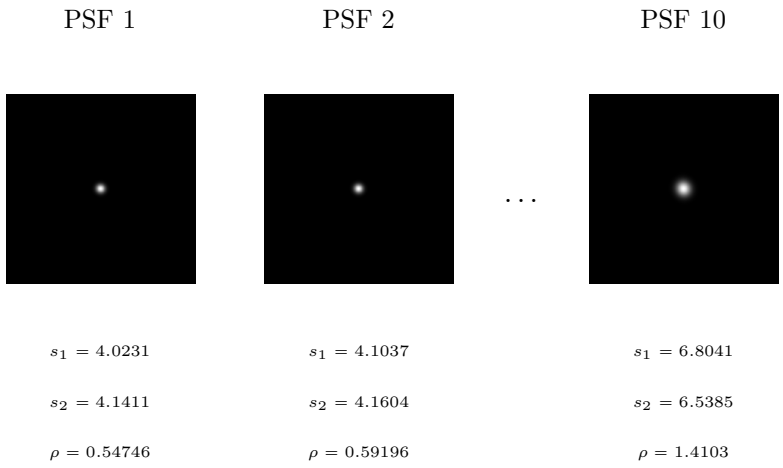


Figure 5.1.: Point spread functions of the model problem, where the values s_1, s_2 and ρ increase from PSF 1 to PSF 10. s_1 and s_2 are random values between 4 and 7. ρ is randomly chosen between 0.5 and 1.5.

indicate the relative error $E_{\mathbf{x}} = \frac{\|\mathbf{x}_{\text{approx}} - \mathbf{x}_{\text{true}}\|_2^2}{\|\mathbf{x}_{\text{true}}\|_2^2}$ of blurred or approximated image with respect to the true image that is displayed in the last column for comparison. Both, relative error and a glance at blurred and deblurred images indicate an improvement in the reconstruction compared with the blurred object for all three approaches. Among the three deblurred images one detects only slight differences.

In addition to image $\mathbf{X}$, the algorithm yields the corresponding approximation to the 10 point spread functions regarding MFBD, and the point spread function of the control frame if we consider the CMF and CSF problem. In Figure 5.3, we illustrate on the left the true point spread function of the first frame, described by PSF 1. As explained above, we choose the initial PSF to be slightly wider than the widest point spread function, PSF 10, that is shown in Figure 5.1. The initial PSF used in these experiments is given next to the true PSF. The right hand side of Figure 5.3 indicates the results for PSF 1 according to the approximation of the true images in Figure 5.2. Above each image, the parameters $\mathbf{y} = (s_1, s_2, \rho)^T$ are stated. As regarding the reconstructed images, the point spread functions are similar for MFBD, CMF and CSF, and represent a good approximation of the true PSF.

In fact, 7 iterations of the Gauss-Newton algorithm lead to the results in Figures 5.2 and 5.3. The upper plot in Figure 5.4 shows the relative error $E_{\mathbf{x}}$ of reconstructed image $\mathbf{X}$ for MFBD, CMF and CSF in each iteration. Below, the relative error for the PSF, $E_{\mathbf{p}} = \frac{\|\mathbf{p}_{\text{approx}} - \mathbf{p}_{\text{true}}\|_2^2}{\|\mathbf{p}_{\text{true}}\|_2^2}$, is illustrated. One observes that the error for both, image and PSF, decreases from iteration to iteration resulting in a stepwise improvement of the reconstructions. The relative error of the reconstructed image nearly matches for MFBD and CMF, and is for each iteration slightly smaller for the CSF problem. On

the other hand, the relative error of the PSF approximation is almost the same for CMF and CSF, but smaller for the classic multiframe blind deconvolution problem. We stop after 7 iterations, since for more iterations the image error increases again.

Experiments With Added Noise Moreover, we apply the Gauss-Newton algorithm with Tikhonov regularization to a data set where 1% of Gaussian white noise is added to the blurred images $\mathbf{B}_1, \ldots, \mathbf{B}_{10}$. The effects can be observed in the deblurred images in Figure 5.5. Especially the CMF approach experiences difficulties with the added noise. Whereas the relative errors of the deblurred images are similar for MFBD and CSF, CMF yields a larger error and a less precise reconstruction. As for the case without noise, we illustrate the reconstructed PSF of the first frame in Figure 5.6 and the relative errors of reconstructed image and point spread function for each iteration in Figure 5.7. In contrast to MFBD and CMF where we perform 7 iterations as in the noise-free case, the relative error of the image reconstructed using the CSF approach increases again after 6 iterations.

Computing Time Figure 5.8 shows two plots comparing the performance of MFBD, CMF and CSF. In the upper figure, the elapsed time per Gauss-Newton iteration is plotted against the number of frames. For MFBD and CMF, the time increases with the number of frames. In contrast, the time for CSF is constant since the size of the least squares problem remains the same. Below, we illustrate the time for seven iterations of the algorithm including for CSF the performance of the Givens rotations. The time for the Givens rotation appears to be negligibly small, since except for the sevenfold time, both figures look similar.

Summarizing, the CSF approach performs best with respect to our model problem. In the noise-free case all three methods yield approximately the same results, whereas CMF experiences problems when noise is added resulting in a less accurate reconstruction than obtained using MFBD or CSF. Moreover, the CSF method has the advantage that its computational costs are lowest compared to MFBD and CMF. However, the presented results leave some questions unanswered. For instance, how the control frame can be determined, what happens if one of the frames 2 to 10 are chosen as control frame, though these frames actually do not satisfy the requirements of being control frame, or if the multiframe approach yields an improvement in the solution compared to the classic blind deconvolution problem. In Section 5.1.1 we regard these questions with respect to the data set used above and three slightly modified data sets.

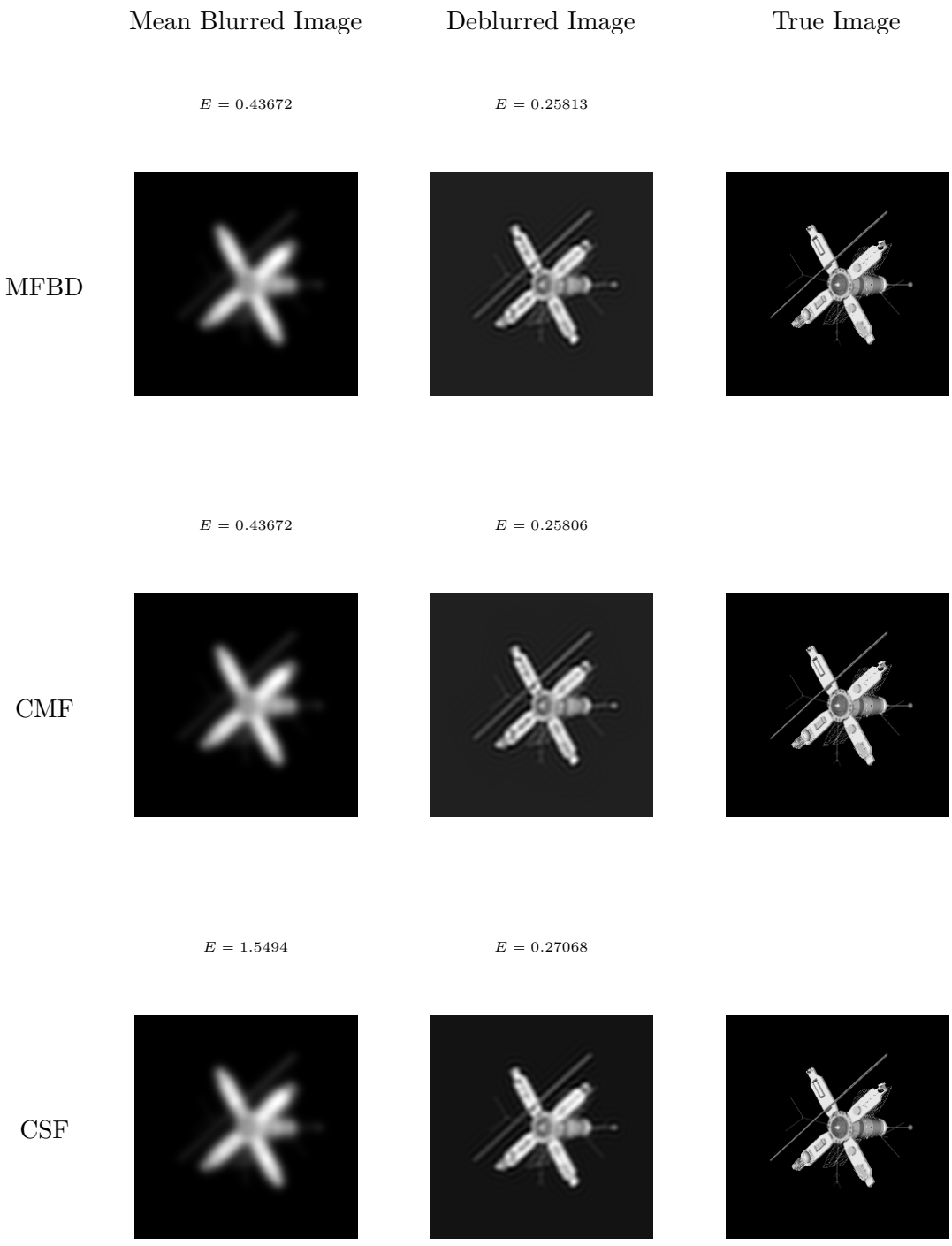


Figure 5.2.: Results for applying 7 iterations of the Gauss-Newton method to the multiframe (MFBD), the compact multiframe (CMF), and the compact single frame deconvolution problem (CSF), respectively. The number of frames is $p = 10$ and we did not add noise to the problem. In fact, the blurred image corresponding to the CSF problem in row 3 shows the blurred image of the control frame after Givens rotation.

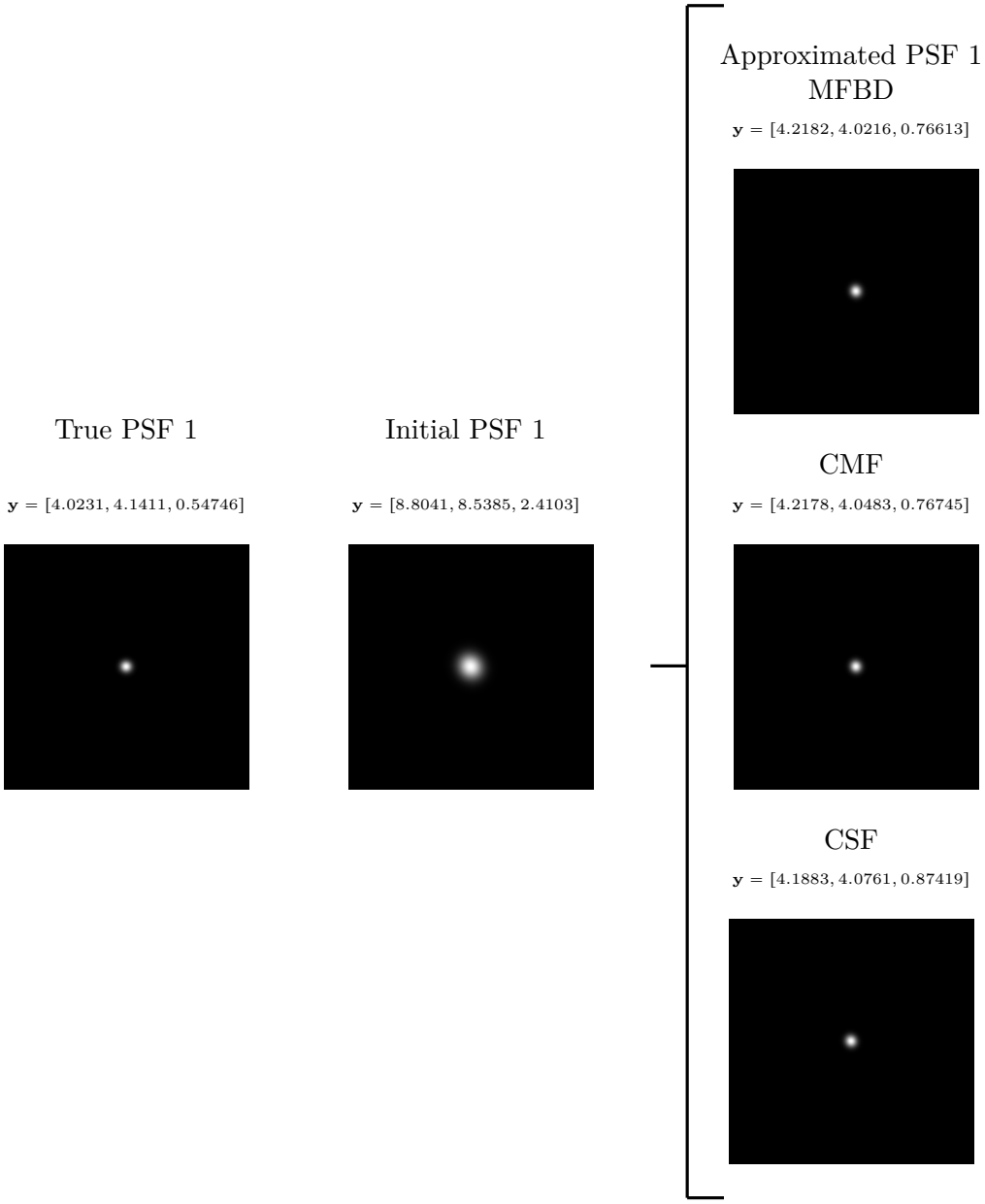


Figure 5.3.: Comparison of reconstructed PSF of frame 1 for MFBD, CMF and CSF. These results correspond to Figure 5.2 (data set with 10 frames, no noise added, 7 iterations of the Gauss-Newton algorithm).

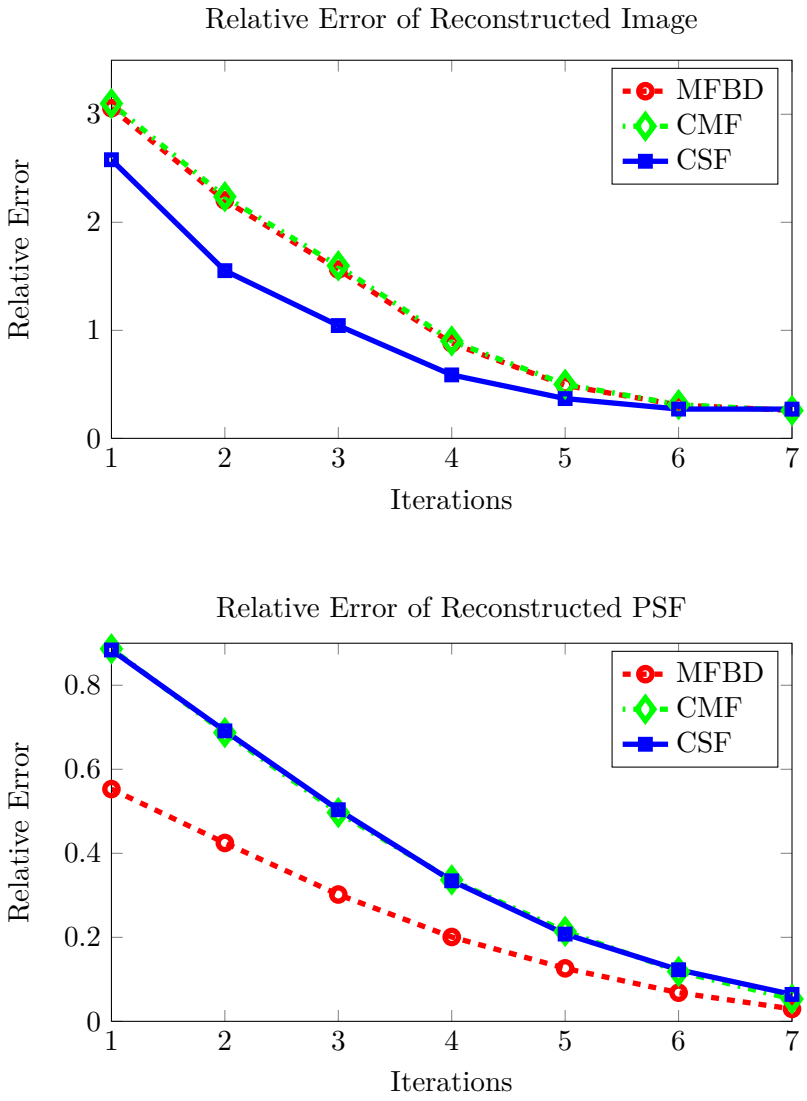


Figure 5.4.: Relative Image and PSF errors corresponding to the results in Figure 5.2.

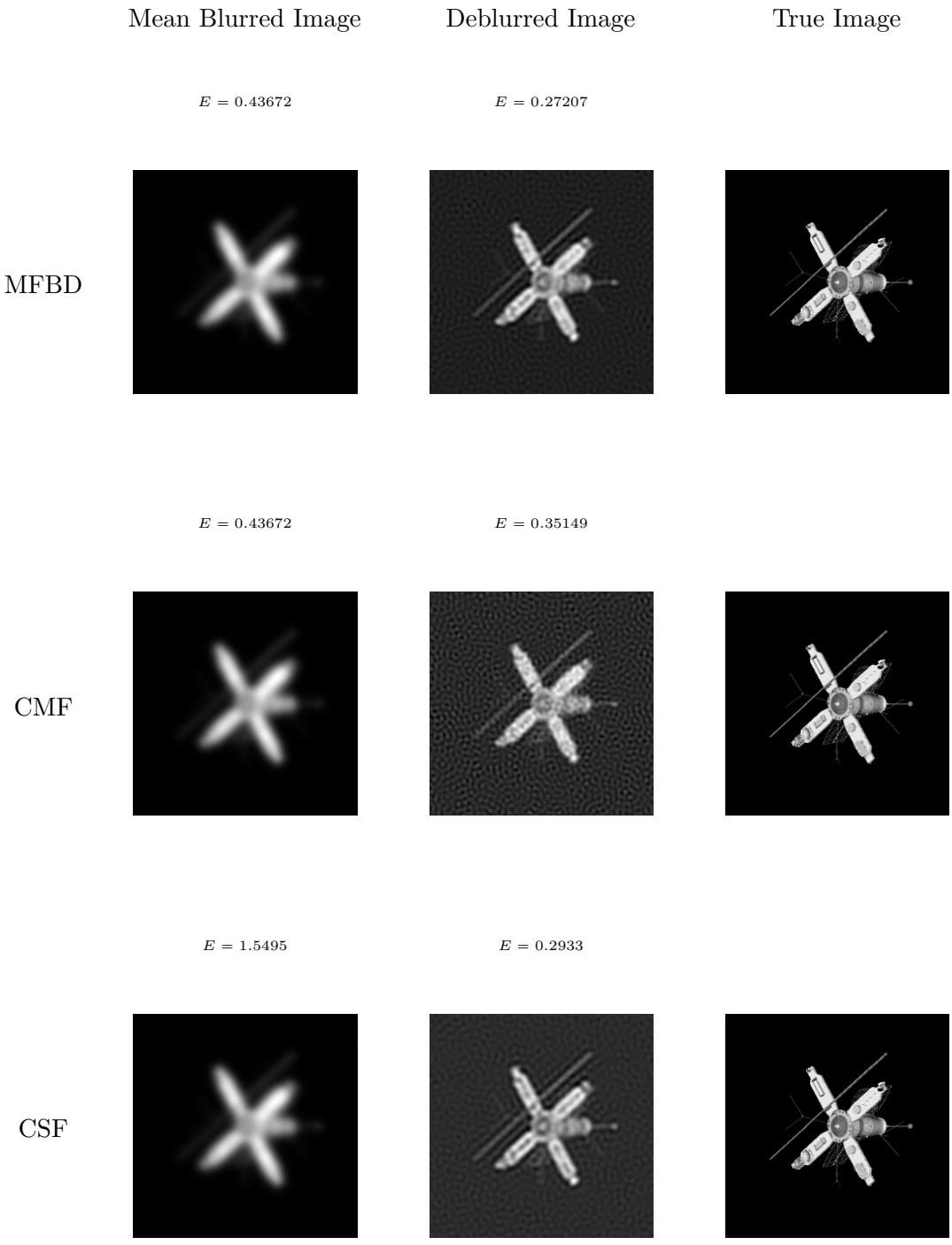


Figure 5.5.: Results for applying the Gauss-Newton method to the MFBD (7 iterations), CMF (7 iterations), and the CSF problem (6 iterations), respectively. The number of frames is $p = 10$ and we added 1% noise to the problem. In fact, the blurred image corresponding to the CSF problem in row 3 shows the blurred image of the control frame after Givens rotation.

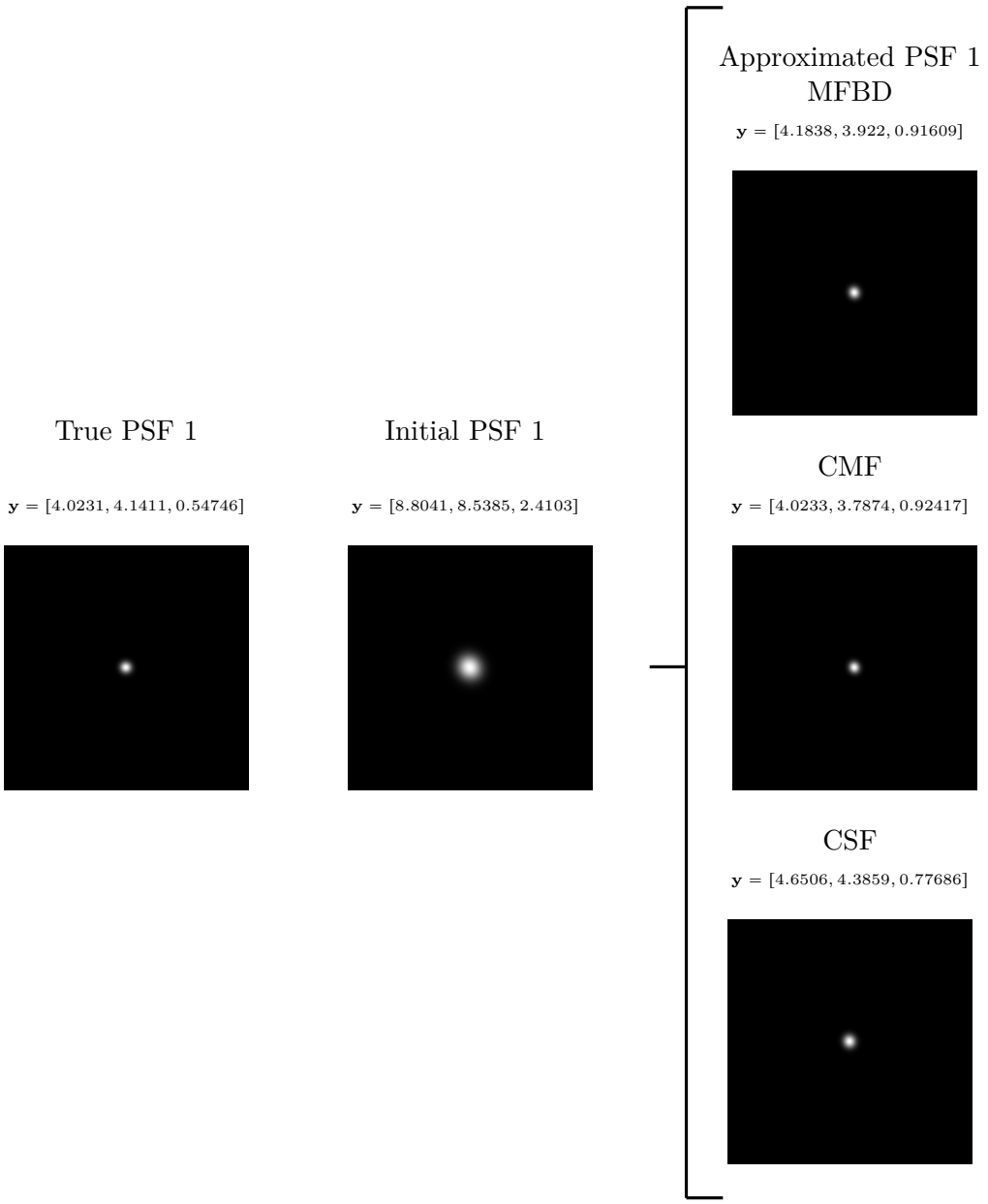


Figure 5.6.: Comparison of reconstructed PSF of frame 1 for MFBD, CMF and CSF. These results correspond to Figure 5.5 (data set with 10 frames, 1% noise added, 7 (MFBD) or 6 (CMF, CSF) iterations of the Gauss-Newton algorithm).

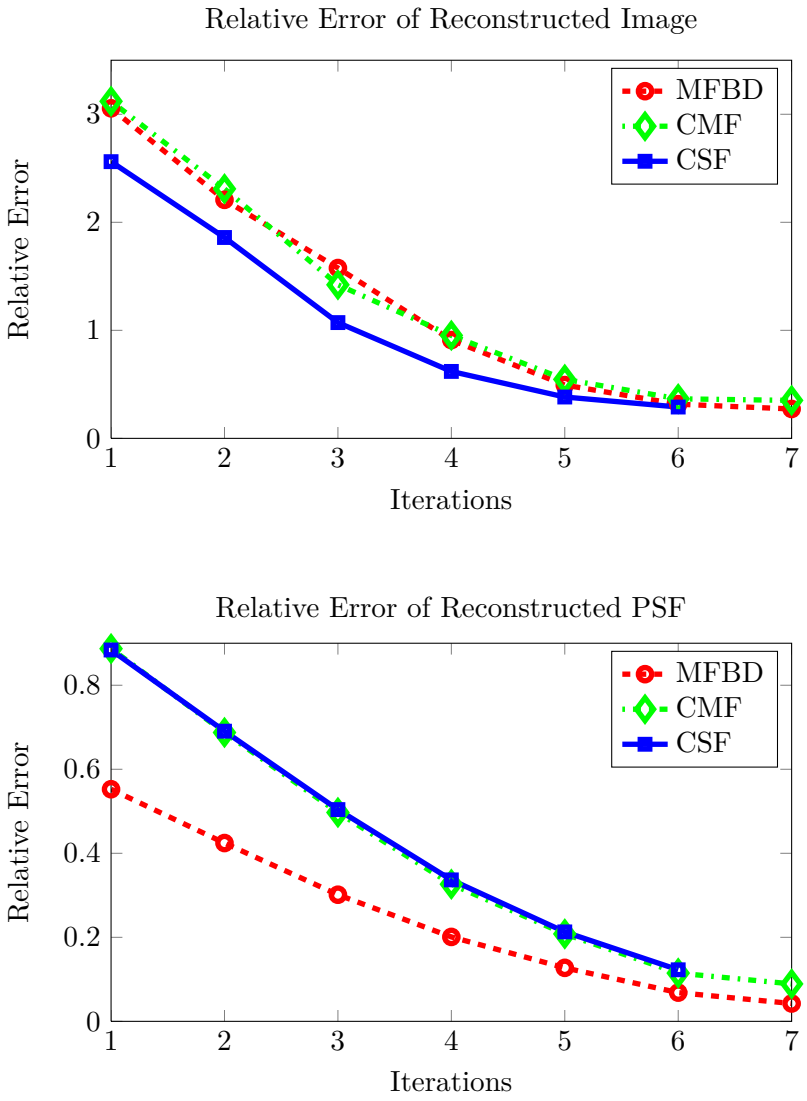


Figure 5.7.: Relative Image and PSF errors corresponding to the results in Figure 5.5.

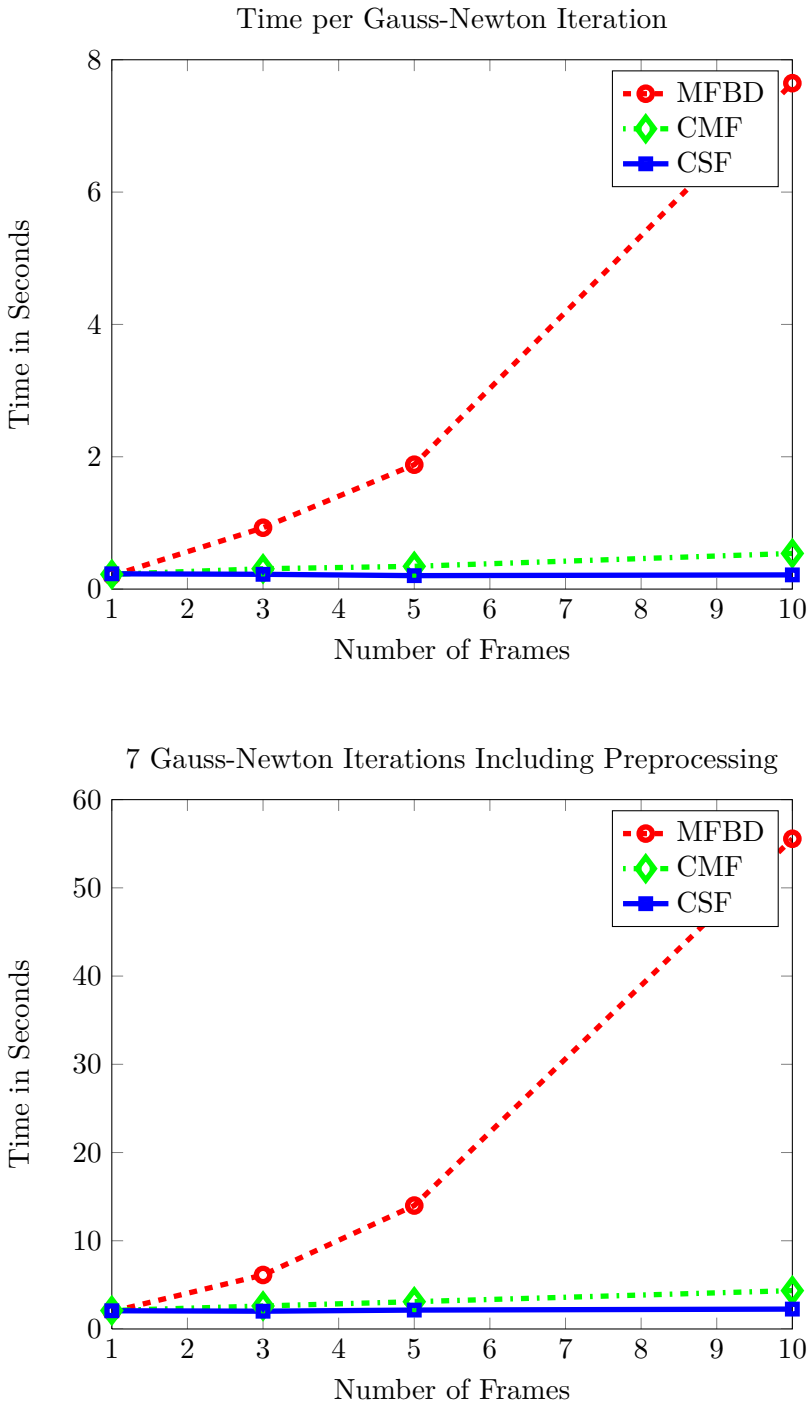


Figure 5.8.: Above, we measured the time it takes to execute one iteration of the algorithm that solves the blind deconvolution problem using the Gauss-Newton method and Tikhonov regularization with GCV (Algorithm 3.2) for the three approaches MFBD, CMF and CSF ignoring runtime for Givens rotation. Below, we determined for the same algorithm the time it takes to perform 7 iterations including preprocessing for CSF where Givens rotation is applied.

5.1.1. Additional Results

We solve the MFBD, CMF and CSF problem using the Gauss-Newton approach for four slightly different model problems, each containing ten frames. True image is the satellite image from the previous section. For each data set, the optimization procedure is performed ten times, each time with a different frame as control frame for CMF and CSF. For comparison, we solve the classic singleframe blind deconvolution problem for frames one to ten.

The results are illustrated in Figures 5.9, 5.10, 5.11 and 5.12. In each figure, the PSFs used to obtain the blurred images are shown in (a). Moreover, we plot the minimum relative error of the reconstructed image for each choice of control frame $(1, \ldots, 10)$ for MFBD, CMF and CSF, respectively. We also plot the minimum relative image error that we receive with singleframe blind deconvolution (BD) of frames one to ten. Subfigure (b) shows these results without noise, whereas in (c) 1% noise is added to the problem.

Figure 5.9 deals with the model problem from above where the true image has been blurred with ten different Gaussian PSFs whose parameters s_1 and s_2 are randomly chosen between 4 and 7, and ρ randomly between 0.5 and 1.5. Moreover, the parameters are sorted such that $s_1^{(1)} < s_1^{(2)} < \ldots < s_1^{(10)}$, $s_2^{(1)} < s_2^{(2)} < \ldots < s_2^{(10)}$ and $\rho^{(1)} < \rho^{(2)} < \ldots < \rho^{(10)}$. Therefore, we choose the first frame as control frame, because its matrix $\mathbf{A}$ has componentwise the least close-to-zero eigenvalues. One observes that solving the BD problem, the image error increases from frame one to ten, since the frames are sorted from least to most blurred. The MFBD error is constant, because it uses all frames. For the problem without added noise, the correct choice of the control frame does not seem important, since CMF and CSF error plots are similar to the MFBD plot. However, adding noise the image error increases from frame one to ten for CMF and CSF.

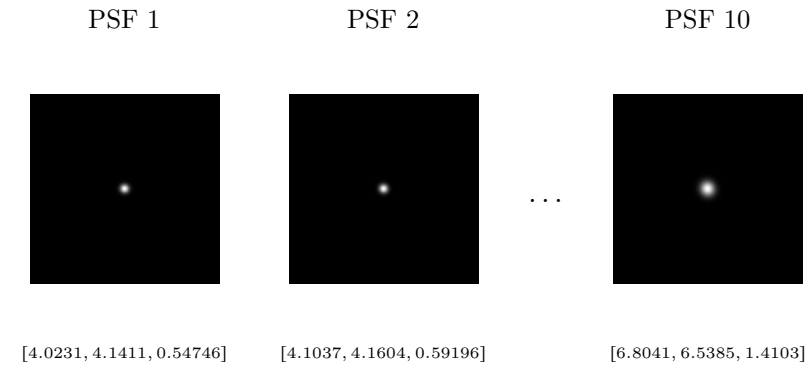
In Figure 5.10 we regard a data set where the PSF parameters s_1 and s_2 are randomly chosen between 4 and 7, and ρ randomly between 0.5 and 1.5. The parameters are unsorted, hence there does not exist a control frame, since none of the PSF matrices $\mathbf{A}$ has componentwise the least close-to-zero eigenvalues. In subfigure (b) where the results without added noise are illustrated, one observes that the minimum errors for MFBD, CMF and CSF do not differ much. Thus, changing the control frame does not result in major changes of the image error. On the other hand, subfigure (c) shows, that adding noise to the problem significantly influences CMF and CSF whose minimal relative errors are relatively large compared to MFBD.

Figure 5.11 considers a data set with ten different Gaussian PSFs whose parameters s_1 and s_2 are randomly chosen between 5 and 6, and ρ randomly between 0.5 and 1.5. As for the first data set, the parameters are sorted such that $s_1^{(1)} < s_1^{(2)} < \ldots < s_1^{(10)}$, $s_2^{(1)} < s_2^{(2)} < \ldots < s_2^{(10)}$ and $\rho^{(1)} < \rho^{(2)} < \ldots < \rho^{(10)}$. Hence following the rule from above, control frame is the first one. The plots behave similar to the plots in Figure

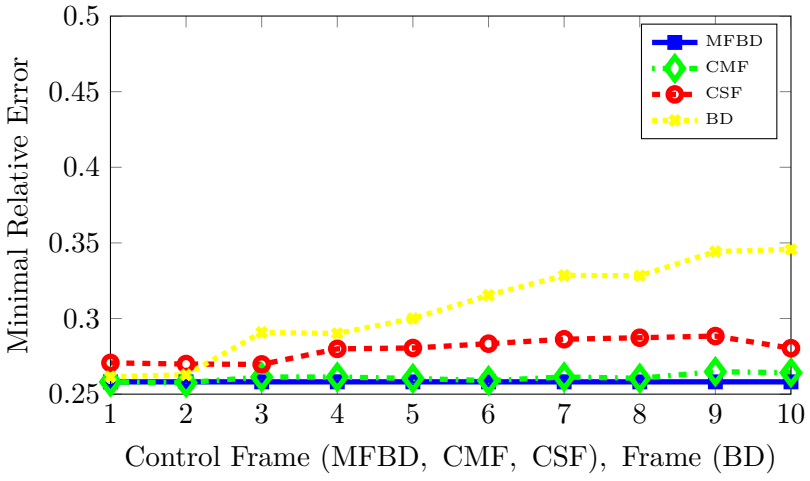
5.9, but narrowing the interval from which we choose the PSF parameters decreases the image error differences. Similar effects can be observed comparing Figure 5.10 to Figure 5.12 where the data set consists of ten different Gaussian PSFs whose (unsorted) parameters s_1 and s_2 are randomly chosen between 5 and 6, and ρ randomly between 0.5 and 1.5.

In a blind deconvolution problem, the PSFs are actually unknown. According to our results, one way to determine the control frame is to choose the frame that belongs to the least blurred image. If the PSFs are very similar, as in the data sets considered in Figures 5.11 and 5.12, the results are also similar and every frame may be chosen as control frame. However, if the frame chosen as control frame does not satisfy the requirements of a control frame, information may be lost which reduces accuracy in the reconstructed image.

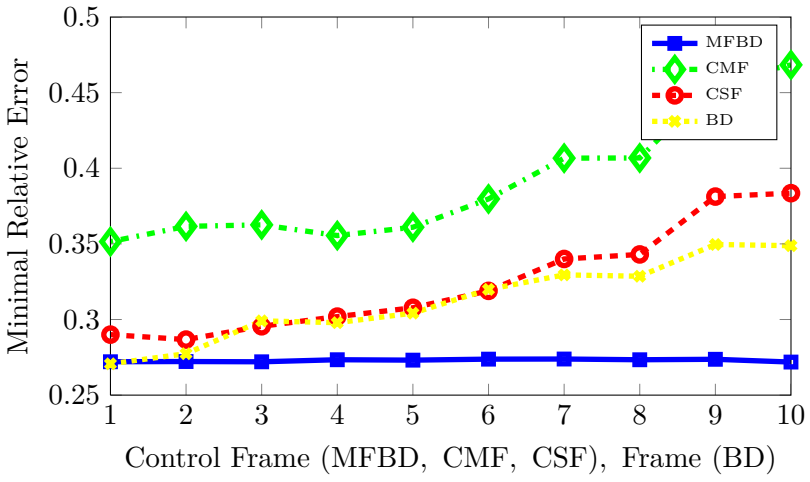
The comparison between the multiframe approaches (MFBD, CMF and CSF) and the classic singleframe approach (BD) shows, that for our simple model, taking the least blurred frame and performing singleframe blind deconvolution can be an advantage. However, if it is not obvious which one to choose, it is advisable to deblur the ten frames simultaneously using multiframe blind deconvolution, instead of trying singleframe blind deconvolution with every of the ten frames.



(a) s_1, s_2 randomly chosen between 4 and 7, ρ randomly chosen between 0.5 and 1.5, and sorted in ascending order. (Equal to Figure 5.1.)

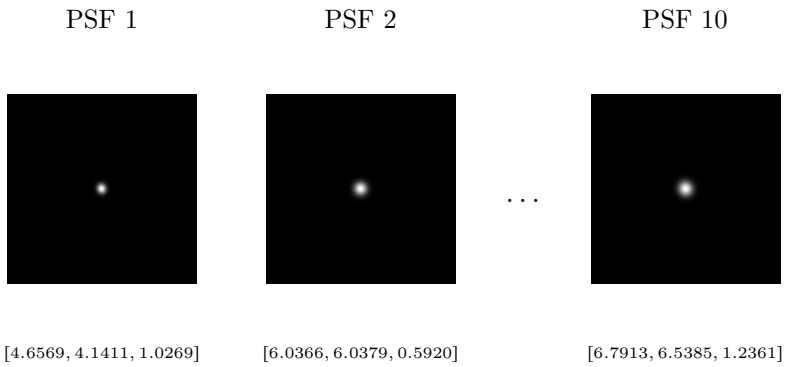


(b) No noise added

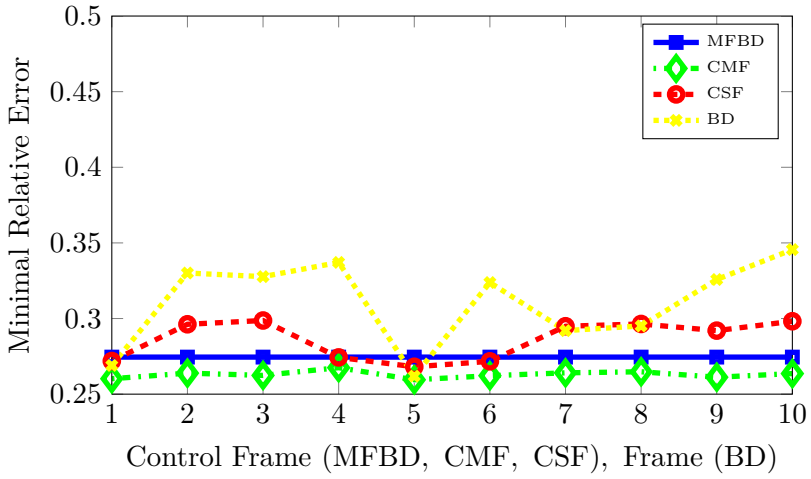


(c) 1% noise added

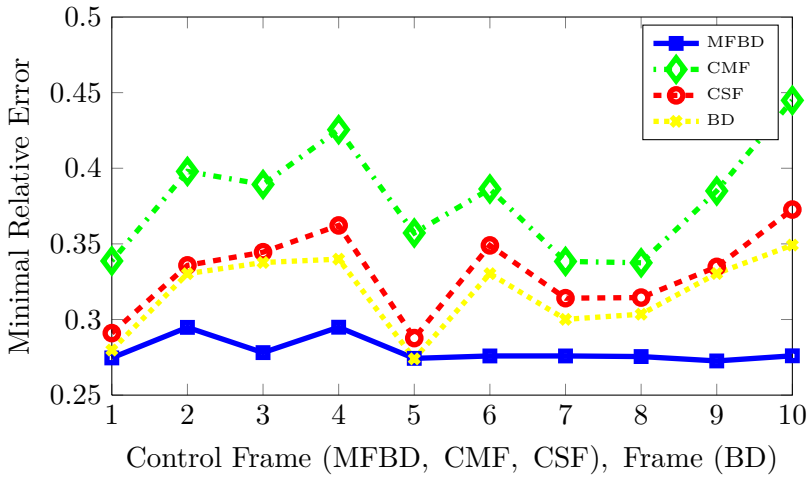
Figure 5.9.: Gauss-Newton approach applied to MFBD, CMF, CSF and BD for the model problem derived from the PSFs in (a). (b) and (c) show the minimum image error plotted against the chosen control frame.



(a) s_1, s_2 randomly chosen between 4 and 7, ρ randomly chosen between 0.5 and 1.5.

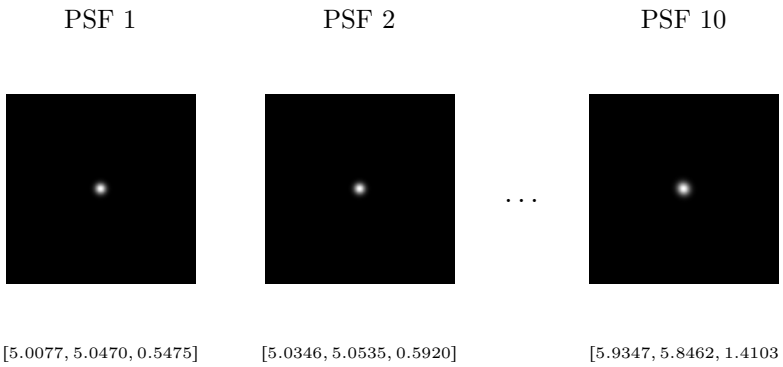


(b) No noise added

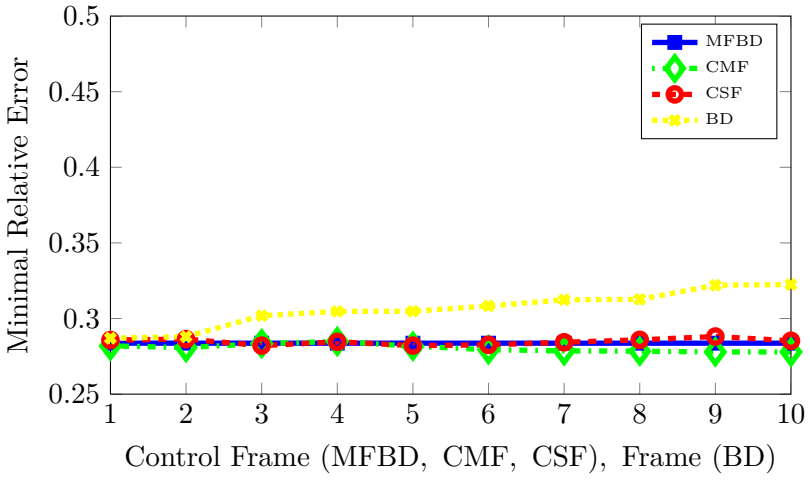


(c) 1% noise added

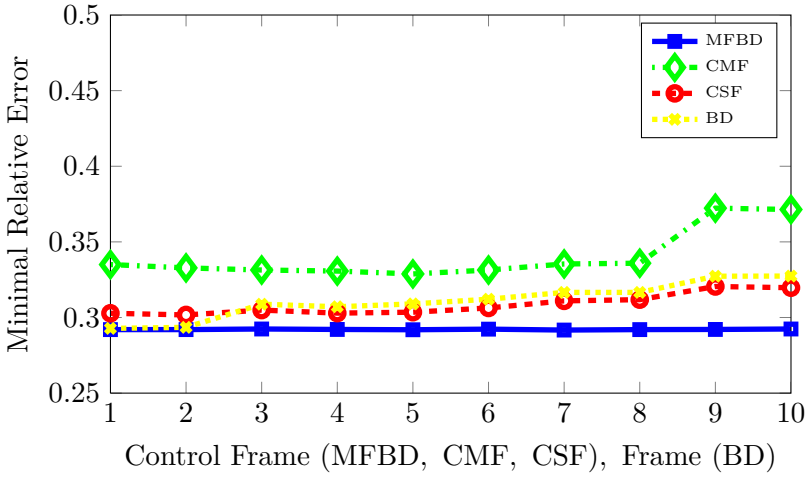
Figure 5.10.: Gauss-Newton approach applied to MFBD, CMF, CSF and BD for the model problem derived from the PSFs in (a). (b) and (c) show the minimum image error plotted against the chosen control frame.



(a) s_1, s_2 randomly chosen between 5 and 6, ρ randomly chosen between 0.5 and 1.5, and sorted in ascending order.

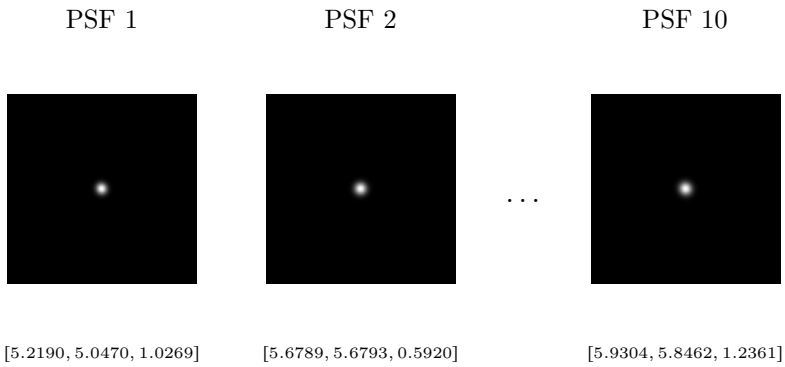


(b) No noise added

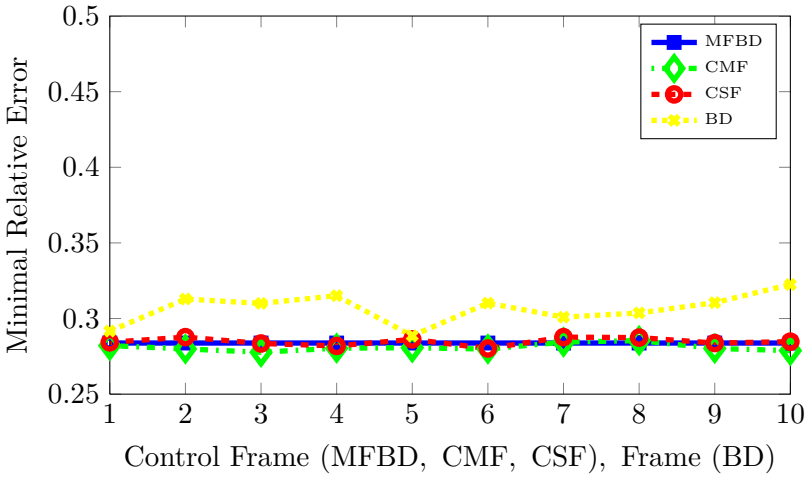


(c) 1% noise added

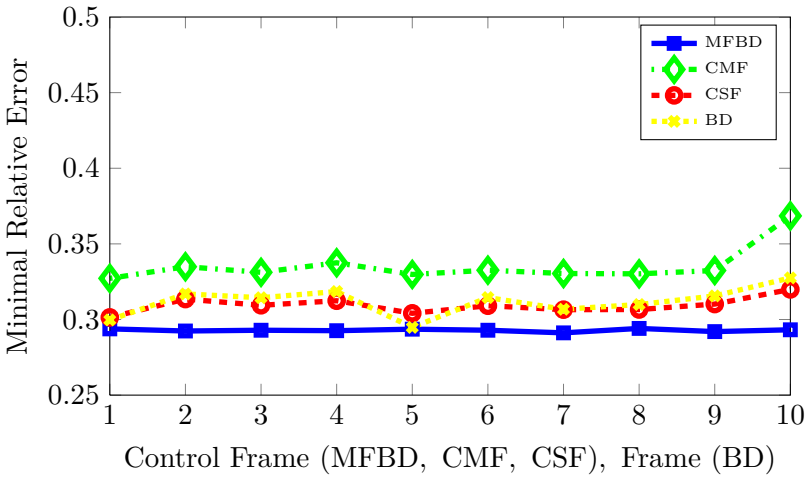
Figure 5.11.: Gauss-Newton approach applied to MFBD, CMF, CSF and BD for the model problem derived from the PSFs in (a). (b) and (c) show the minimum image error plotted against the chosen control frame.



(a) s_1, s_2 randomly chosen between 5 and 6, ρ randomly chosen between 0.5 and 1.5.



(b) No noise added



(c) 1% noise added

Figure 5.12.: Gauss-Newton approach applied to MFBD, CMF, CSF and BD for the model problem derived from the PSFs in (a). (b) and (c) show the minimum image error plotted against the chosen control frame.

5.2. Direct Approach: APEX Method

In comparison to the iterative approach using Gauss-Newton, we solve the CSF optimization problem (4.8) directly. Therefore, we approximate the PSF of the control frame by means of the APEX method introduced in Section 3.1.1, and reconstruct the sharp image using Tikhonov regularization with GCV.

Again, we design a model problem consisting of ten frames of size 256×256 blurring a sharp image $\mathbf{X}$ with ten Gaussian PSFs of different width obtaining ten varyingly blurred images $\mathbf{B}_1, \ldots, \mathbf{B}_{10}$. Here, the width of the point spread functions increases from frame to frame, that is $0 < \alpha_1^{(1)} < \alpha_1^{(2)} < \ldots < \alpha_1^{(10)}$ and $0 < \alpha_2^{(1)} < \alpha_2^{(2)} < \ldots < \alpha_2^{(10)}$, such that the first one is the control frame. The point spread functions used to obtain the data set of blurred images are shown in Figure 5.13.

Throughout this thesis, we came across two different expressions for Gaussian PSFs. First, the Gaussian function described in Section 2.2 with parameters s_1, s_2 and ρ, and second, the Fourier transform of a Gaussian PSF as a special case of Lévy stable PSFs with parameters α_1 and α_2 given in (3.12). As explained in 3.1.1, the APEX method uses this second expression. To derive the Gaussian PSF from its Fourier transform, we apply MATLAB's `ifft2` to the Fourier transform representation and scale the result such that the sum of all entries is equal to one. Since the second Gaussian PSF version cannot be modified with respect to its orientation, we set $\rho = 0$ in the first Gaussian PSF version for a better comparison between both functions. From this we deduce that $\alpha = \frac{2s^2\pi^2}{mn}$ for a PSF array of size $m \times n$. Hence, the point spread functions in Figure 5.13 most closely approximate the point spread functions we use to build the model problem for the Gauss-Newton experiments illustrated in Figure 5.1, except for the orientation given by ρ. Next, we set up the CMF problem (4.6) and transfer it into CSF expression (4.8) applying Givens rotations. Seeking to reconstruct both unknowns, PSF of the control frame and true image, we first apply the APEX method to blurred image $\mathbf{B}_1$ to estimate the PSF. Figure 5.14 illustrates logarithmized mid-row and mid-column of the image's Fourier transform. According to Section 3.1.1, we only consider the region of both plots in which they lie above noise level. In this case we cut out intervals $[-82, 82]$ and $[-80, 80]$, respectively. In contrast to the example in Section 3.1.1, Figure 3.4, we consider a Gaussian PSF with two different parameters α_1 and α_2. Thus we have to look at both horizontal and vertical ray, as explained in Section 3.1.1.

Figure 5.15 shows a picture of the MATLAB GUI used to approximate both plots by a function $f(x) = -\alpha x^2 - \mathscr{A}$ where $\mathscr{A}$ can be adjusted. Subsequently, the corresponding PSF and deblurred image are computed. We discover that the formation of artifacts in the deblurred image decreases with increasing apex value $\mathscr{A}$. However, reducing artifacts by choosing large values for $\mathscr{A}$ is accompanied by an increasing smoothness. Therefore, we have to find a balance between artifacts and blur.

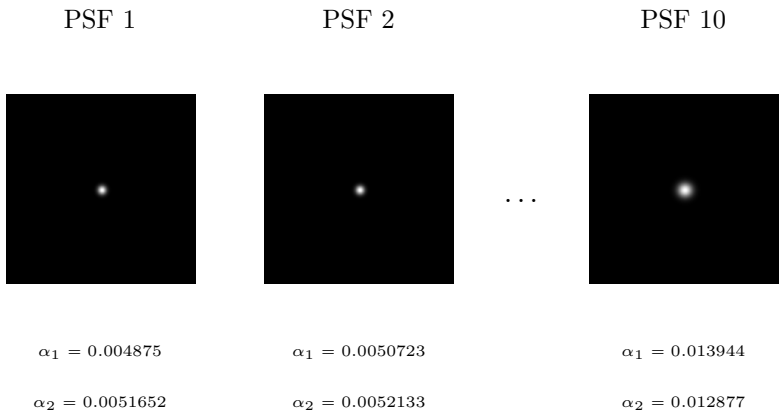


Figure 5.13.: Point spread functions of the multiframe model problem for experiments using the APEX Method. α_1 and α_2 increase from PSF 1 to PSF 10. The parameters are random values between 0.0048 and 0.0148

The error plot illustrated in Figure 5.16 shows relative image and PSF errors for apex values between 0 and 10. Though the image error is minimal at around $\mathscr{A} = 2.5$, the deblurred image shows severe artifacts. Also for $\mathscr{A} = 4.5$ where the PSF error roughly attains its minimum, the reconstruction still exhibits perturbations along the satellite's borders. A transition between a perturbed and a smoothed image can be observed at around $\mathscr{A} = 6.5$.

In Figure 5.17, we compare true and approximated point spread functions for $\mathscr{A} = 4.5$ and $\mathscr{A} = 6.5$. The corresponding deblurred images are illustrated in Figure 5.18.

Experiments With Added Noise Furthermore, we examine the reconstruction of point spread function and true image for a data set with 1% Gaussian white noise added to the blurred images $\mathbf{B}_1, \ldots, \mathbf{B}_{10}$. As above, Figure 5.19 indicates horizontal and vertical rays through the center of $\log|\hat{\mathbf{B}}_1|$. One observes that the intervals for which the plots lie above noise level, in particular $[-23, 23]$ and $[-27, 27]$, are much smaller than for the noise-free case.

The MATLAB GUI in Figure 5.20 shows how only a fraction of both plots is used to reconstruct the point spread function. For an apex value $\mathscr{A}$ above 5, the approximating curve $f(x) = -\alpha x^2 - \mathscr{A}$ becomes a horizontal line leading to impractical reconstructions.

Apex value $\mathscr{A} = 3$ yields the best results, which is supported by the error plots in Figure 5.21. Figure 5.22 shows the reconstructed PSF and compares it to the true PSF. We use the PSF approximation to compute a reconstruction of the true image illustrated in Figure 5.23.

In conclusion, we can see that image deblurring using the APEX method can also work well if the blurry images are perturbed by a certain amount of noise.

Computing Time and Comments As examined above in Section 5.1 presenting experiments and results using the Gauss-Newton method, the Givens-rotation step can be computed in a fraction of a second for an image of size 256×256. User-interaction is required by choosing an appropriate interval above noise level regarding the plots in Figure 5.14 and 5.19. Furthermore, the MATLAB GUI involves a certain amount of interactivity. Each curve fit and the subsequent deblurring are computed instantaneously. However, trying to find the perfect apex value can be difficult and time-consuming.

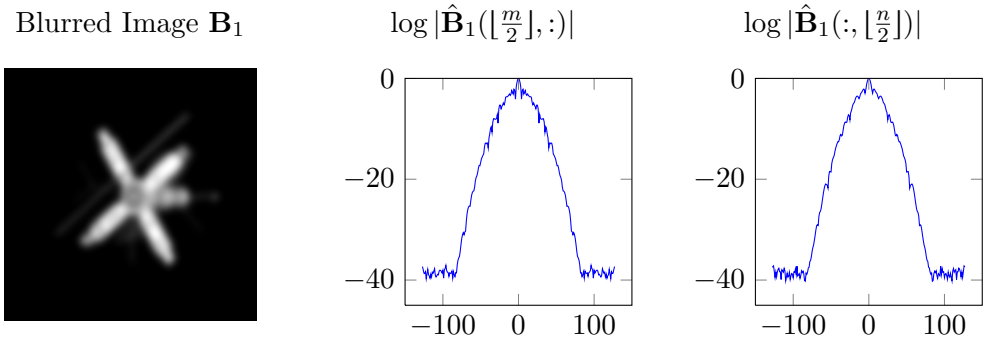


Figure 5.14.: Blurred image of frame 1 and corresponding horizontal and vertical rays through its logarithmized Fourier transform.

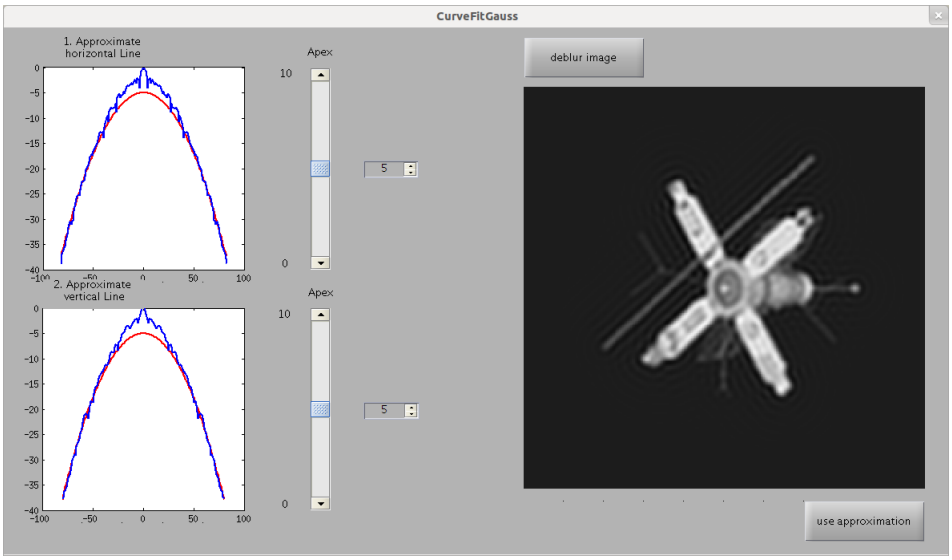


Figure 5.15.: MATLAB GUI

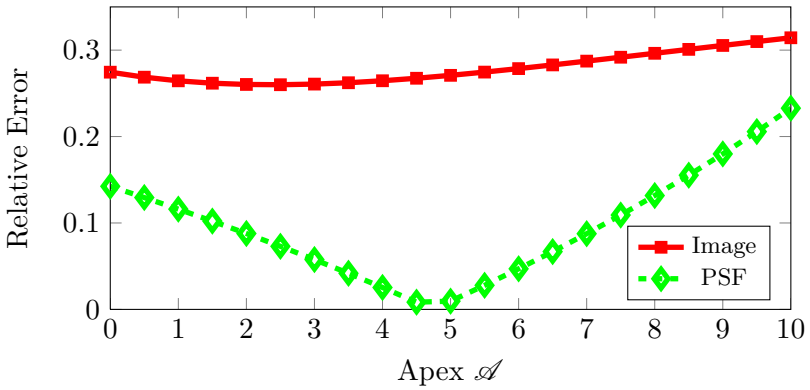


Figure 5.16.: Relative error for reconstructed PSF and corresponding image for different apex values.

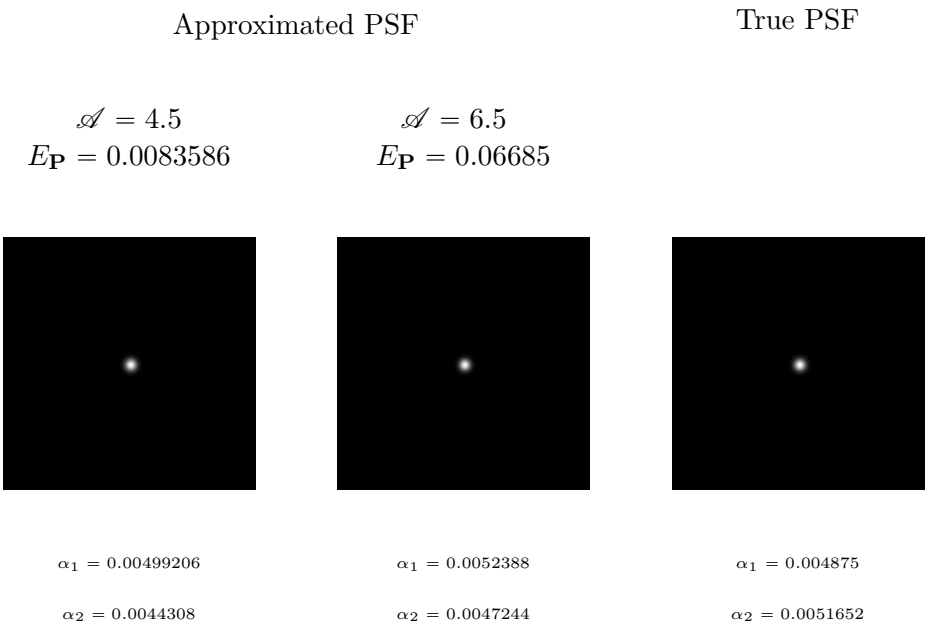
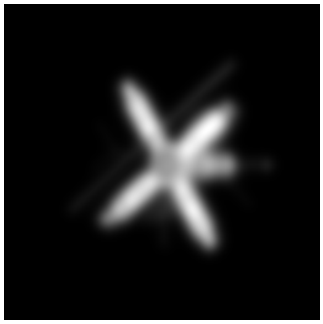


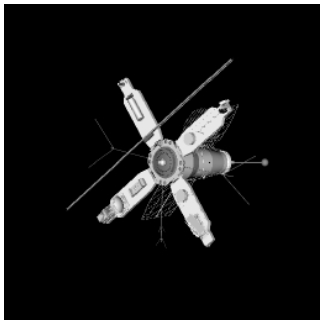
Figure 5.17.: PSFs approximated using APEX Method with apex values $\mathscr{A} = 4.5$ and $\mathscr{A} = 6.5$, respectively.

Blurred Image $\tilde{\mathbf{B}}_1$

$E_{\mathbf{X}} = 1.5784$



True Image $\mathbf{X}$



Deblurred Image

$\mathscr{A} = 4.5$
$E_{\mathbf{X}} = 0.26745$

Deblurred Image

$\mathscr{A} = 6.5$
$E_{\mathbf{X}} = 0.28288$

Figure 5.18.: Reconstructed images using PSFs approximated with APEX method for deblurring. The corresponding PSFs are illustrated in Figure 5.17.

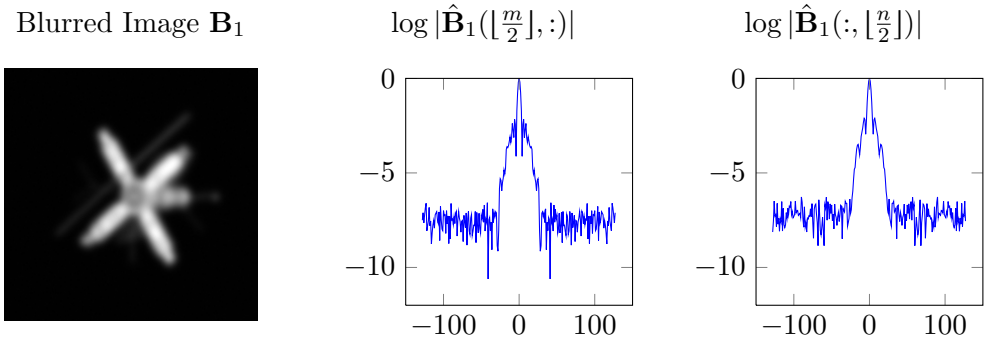


Figure 5.19.: Blurred image of frame 1 and corresponding horizontal and vertical rays through its logarithmized Fourier transform (Noise).

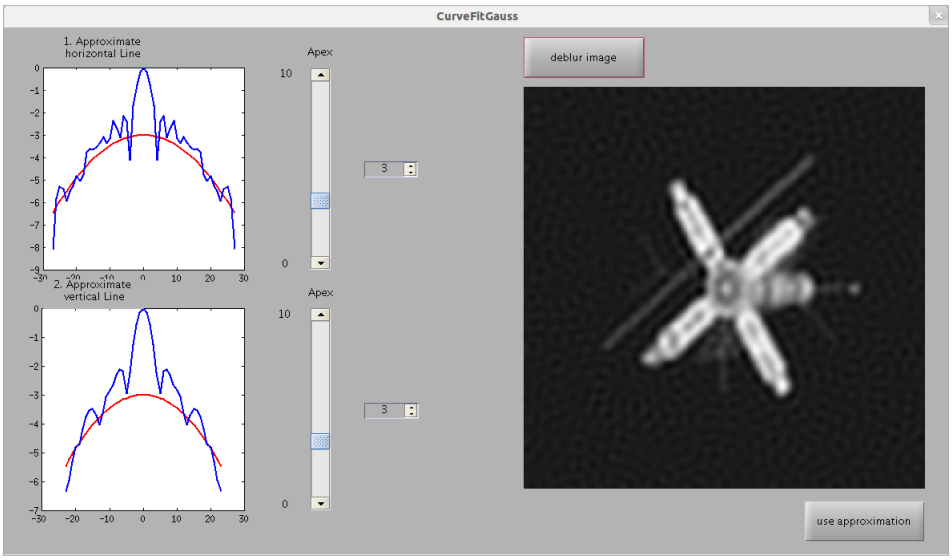


Figure 5.20.: MATLAB GUI (Added Noise)

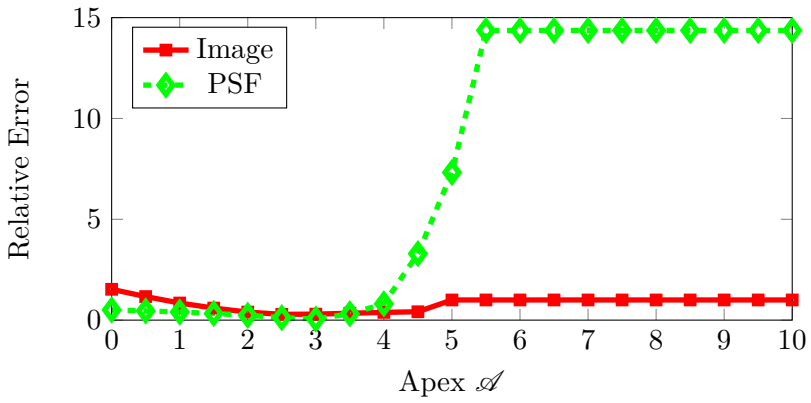


Figure 5.21.: Relative error for reconstructed PSF and corresponding image for different apex values (Noise).

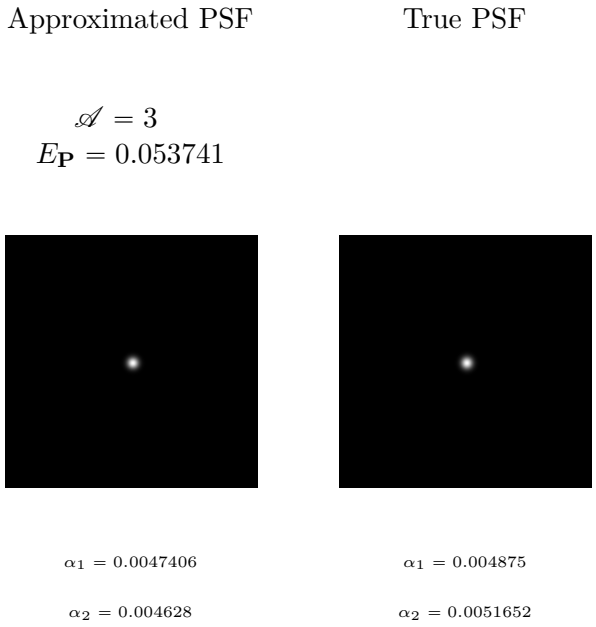
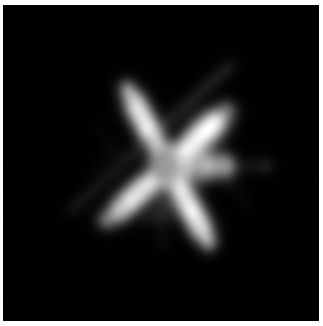


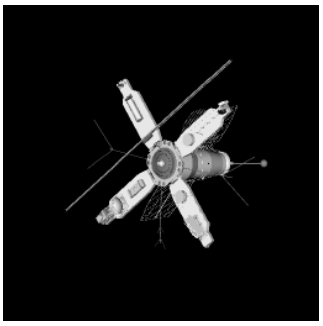
Figure 5.22.: PSFs approximated using APEX Method with apex values $\mathscr{A} = 3$ (Noise).

Blurred Image $\tilde{\mathbf{B}}_1$

$E_{\mathbf{X}} = 1.5784$



True Image $\mathbf{X}$



Deblurred Image

$\mathscr{A} = 3$
$E_{\mathbf{X}} = 0.29773$

Figure 5.23.: Reconstructed images using PSFs approximated with APEX method for deblurring. The corresponding PSFs are illustrated in Figure 5.22.

6. Summary, Conclusion, and Outlook

This work has reviewed efficient direct and iterative approaches solving the blind deconvolution problem, transferred these concepts to the multiframe blind deconvolution problem, and examined and developed methods simplifying the multiframe blind deconvolution problem to significantly accelerate computations.

For blind deconvolution, we have presented the iterative alternating least squares and Gauss-Newton methods, as well as a direct strategy using the APEX method to estimate the point spread function. In each case, the Tikhonov method with GCV provides a regularized solution, and we impose periodic boundary conditions allowing fast computations in the Fourier domain.

The Gauss-Newton method requires an initial guess for the parameters defining the point spread function, which is difficult to figure out in practice. Here, we described the algorithm in the case of Gaussian PSFs with parameters s_1, s_2 and ρ, but it can be similarly applied to problems where the blur is caused by other PSFs. However, it is necessary to know which parameters the point spread function depends on and how.

Unfortunately, applying the APEX method involves limitations regarding the PSF. For Gaussian PSFs, a special case of Lévy stable PSFs, we derived a method to reconstruct doubly symmetric PSFs, but in general, only rotationally symmetric Lévy stable PSFs can be obtained. Furthermore, the APEX method requires user interaction, which takes up additional time. On the other hand, regarding a test problem where all these requirements are met, we receive very good results and computation times are short. Another advantage is, that the only input taken by the algorithm is the blurred image. An initial guess for point spread function or true image is not necessary.

Considering multiframe blind deconvolution instead of blind deconvolution, the underdeterminedness of the problem can be slightly decreased. We solved the problem using Gauss-Newton with Tikhonov regularization. The problem size increases with the number of frames, and so does the computation time. Likewise, the more frames a problem counts, the less advisable it is to approximate every frame's PSF with the APEX method, since manually selecting an apex value for each frame is a time-consuming procedure.

Given that a control frame exists, the multiframe blind deconvolution problem can be simplified to *compact* multiframe blind deconvolution, depending only on the PSF of the control frame. We solved the compact problem using Gauss-Newton as well

as APEX. Gauss-Newton iterations are performed significantly faster than before, but time still increases with the number of frames. The APEX method is also a good choice, since only one PSF has to be estimated. Applying Givens rotations, we reduced the compact multiframe blind deconvolution to a singleframe problem. Due to decreased problem size, computations could be further accelerated. All experiments with artificial data yielded satisfactory results.

The presented approaches show several areas for future development. First, user-friendliness and speed of the APEX method can be improved by automating the interactive parameter choices. Moreover, the transfer from multiframe to compact multiframe blind deconvolution shows possibilities for further development. In cases where no single control frame can be detected, we might consider a combination of frames. In addition, it would be interesting to substitute our version of the Tikhonov method for another regularized solving method that does not explicitly solve normal equations to improve stability of numerical computations. Furthermore, we assumed the image to be periodically repeated beyond its border, which leads to artifacts in cases where objects or patterns close to the borders are not similar to opposite borders. This, however, does not often occur in real image data. Reflexive boundary conditions, on the other hand, are better suited for images with objects near the boundaries. Hence, adapting the methods to be used with reflexive boundary conditions instead of periodic boundary conditions and comparing the results might also be worth investigating. Finally, it would be interesting to run tests with real image data.

A. Appendix

A.1. List of Symbols and Abbreviations

Image Symbols

PSF	Point spread function
B	Blurred image
X	True image
P	PSF
$s_1, s_2, \rho \in \mathbb{R}$	Gaussian PSF parameters (width and orientation)
(k, l)	Center of PSF
$\mathbf{y}$	Column vector containing parameters of PSF, e.g. s_1, s_2, and ρ
$\mathbf{B}, \mathbf{X}, \mathbf{P} \in \mathbb{R}_+^{m \times n}$	Array representation of B, X, and P
$\hat{\mathbf{B}}, \hat{\mathbf{X}}, \hat{\mathbf{P}} \in \mathbb{C}^{m \times n}$	Fourier transform of $\mathbf{B}$, $\mathbf{X}$, and $\mathbf{P}$
$\mathbf{b}, \mathbf{x}, \mathbf{p} \in \mathbb{R}_+^{mn \times 1}$	Vector representation of $\mathbf{B}, \mathbf{X},$ and $\mathbf{P}$
$\hat{\mathbf{b}}, \hat{\mathbf{x}}, \hat{\mathbf{p}} \in \mathbb{C}^{mn \times 1}$	Vector representation of $\hat{\mathbf{B}}, \hat{\mathbf{X}}, \hat{\mathbf{P}}$
$\mathbf{P}_{\curvearrowleft}$	$\mathbf{P}$ rotated by 180°
$\mathbf{P}_{\mathsf{circ}}$	Circshifted array $\mathbf{P}$
$\mathbf{X}_{\mathsf{BC}}$	Expanded $\mathbf{X}$, e.g. with zeros, circularly or reflexive
$\mathbf{b}_{\mathsf{exact}} \in \mathbb{R}_+^{mn \times 1}$	Blurred image without noise in vector representation
$\mathbf{e} \in \mathbb{R}^{mn \times 1}$	Error, noise

Linear Algebra Symbols

$\mathbf{1}_N$	$N \times 1$ vector of ones
$\mathbf{I}_N$	$N \times N$ identity matrix
v_i	Element i of vector $\mathbf{v}$
m_{ij}	Element (i, j) of matrix $\mathbf{M}$
$\mathbf{m}_i$	Column i of matrix $\mathbf{M}$

Matrix and Array Operations

$\mathrm{vec}(\cdot)$	Vector representation (stacking columns of array)
$\mathrm{diag}(\cdot)$	If input is matrix: diagonal of a matrix; if input is vector: diagonal matrix with vector on diagonal
$\mathrm{tr}(\cdot)$	Trace of a matrix
$\mathbf{M}^*$	Complex conjugate transpose of a matrix $\mathbf{M}$
$\mathbf{M}^+$	Pseudoinverse of a matrix
$\mathbf{M}_1 \otimes \mathbf{M}_2$	Kronecker product of two matrices

Special Matrices

BTTB	Block Toeplitz matrix with Toeplitz blocks
BTHB	Block Toeplitz matrix with Hankel blocks
BHTB	Block Hankel matrix with Toeplitz blocks
BHHB	Block Hankel matrix with Hankel blocks
BCCB	Block circulant matrix with circulant blocks
$\mathbf{A} \in \mathbb{R}_+^{mn \times mn}$	PSF matrix

$\mathbf{T}$	$N \times N$ Toeplitz matrix
$\mathbf{C}$	$N \times N$ circulant matrix
$\mathbf{H}$	$N \times N$ Hankel matrix
$\mathbf{\Lambda_C}$	Diagonal matrix of eigenvalues $\lambda_i, i = 1, \ldots, N$ of $\mathbf{C}$
$\mathbf{\Lambda}, \mathbf{\Lambda_A}$	Diagonal matrix of eigenvalues $\lambda_i, i = 1, \ldots, mn$ of $\mathbf{A}$
$\mathbf{U}$	Matrix containing left singular vectors $\mathbf{u}_i$ of $\mathbf{A}$
$\mathbf{V}$	Matrix containing right singular vectors $\mathbf{v}_i$ of $\mathbf{A}$
$\mathbf{\Sigma}$	Diagonal matrix of singular values $\sigma_i, i = 1, \ldots, mn$ of $\mathbf{A}$
$\mathbf{D} \in \mathbb{C}^{mn \times mn}$	Diagonal matrix

Fourier and Cosine Transform

DFT	Discrete Fourier transform
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform
$\mathbf{F}_N, \mathbf{F}_N^*$	1D DFT matrix and inverse DFT matrix of size $N \times N$
$\mathbf{F} = \mathbf{F}_n \otimes \mathbf{F}_m, \mathbf{F}^*$	2D DFT matrix and inverse DFT matrix of size $mn \times mn$
$\mathcal{F}$	Continuous Fourier transform
$\mathbf{K}$	DCT matrix

Regularization

GCV	Generalized cross-validation
ϵ	Regularization parameter
$\mathbf{\Phi}$	Diagonal matrix where $\phi_{ii} = \frac{\sigma_i^2}{\sigma_i^2 + \epsilon}$

APEX Method

$\hat{P}_{\alpha,\beta}$	Fourier transform of Lévy stable PSF ($\alpha > 0, 0 < \beta \leqslant 1$)
$\hat{\mathbf{X}}^{\gamma}, \hat{\mathbf{B}}^{\gamma}$	Normalized Fourier transform of $\mathbf{X}, \mathbf{B}$
$\log(\hat{\mathbf{X}}^{\gamma}), \log(\hat{\mathbf{B}}^{\gamma})$	Element-wise logarithm of $\hat{\mathbf{X}}^{\gamma}, \hat{\mathbf{B}}^{\gamma}$
$\boldsymbol{\xi} \in \mathbb{N}^{1\times n}$	Vector containing the equidistant values between $-\lceil \frac{n}{2} \rceil + 1, \ldots, \lfloor \frac{n}{2} \rfloor$
$\boldsymbol{\eta} \in \mathbb{N}^{1\times m}$	Vector containing the equidistant values between $-\lceil \frac{m}{2} \rceil + 1, \ldots, \lfloor \frac{m}{2} \rfloor$
$\boldsymbol{\Xi} \in \mathbb{N}^{n\times m}$	Matrix where every row is vector $\boldsymbol{\xi}$
$\mathbf{H} \in \mathbb{N}^{n\times m}$	Matrix where every column is vector $\boldsymbol{\eta}$
$\mathscr{A}, \mathcal{A}$	Apex value, and vector containing apex value at every position

Iterative Methods

ALS	Alternating least squares
$\mathbf{r}$	Residual
$\mathbf{J_r}$	Jacobian matrix of $\mathbf{r}$
$\mathbf{d}$	Descent direction
a	Step length

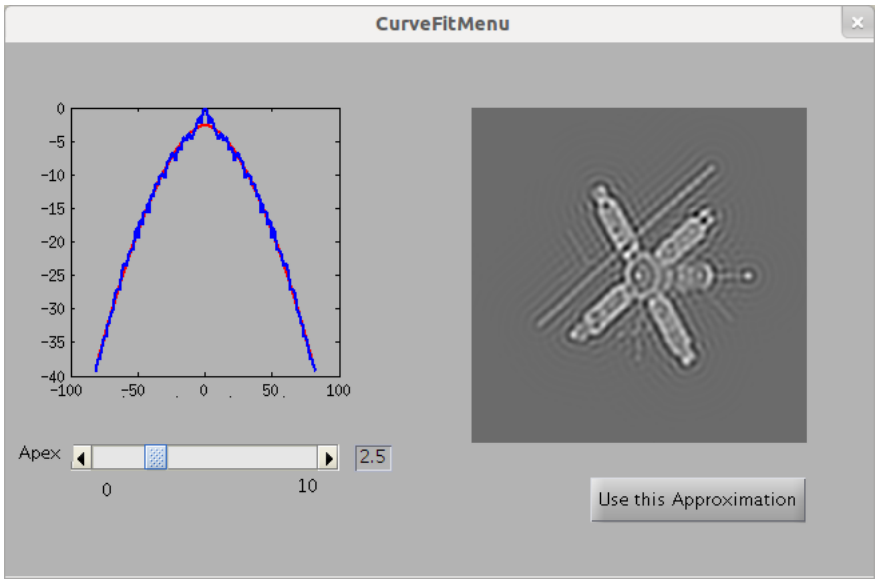
Givens Rotation

$\mathbf{G}$	Givens rotation
$\mathbf{Q}$	Product of Givens rotations $\mathbf{G}_1, \ldots, \mathbf{G}_n$
$\tilde{\mathbf{b}}$	Givens rotated vector $\mathbf{b}$

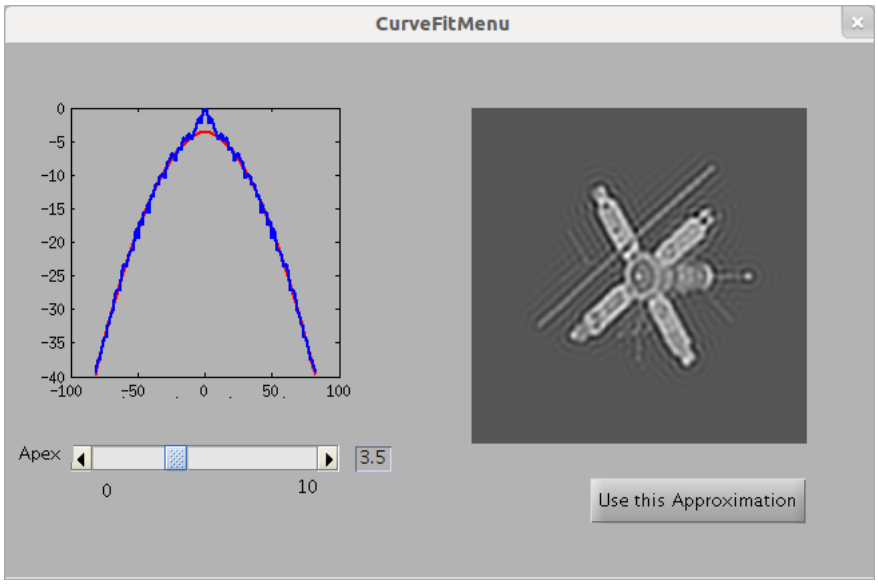
Other Symbols and Abbreviations

CCD	Charge-coupled device
LBT	Large Binocular Telescope
MFBD	Multiframe blind deconvolution
CMF	Compact multiframe blind deconvolution
CSF	Compact single frame blind deconvolution
$E_{\mathbf{x}}$	Relative error of true image $\mathbf{X}$
$E_{\mathbf{p}}$	Relative error of true PSF $\mathbf{P}$

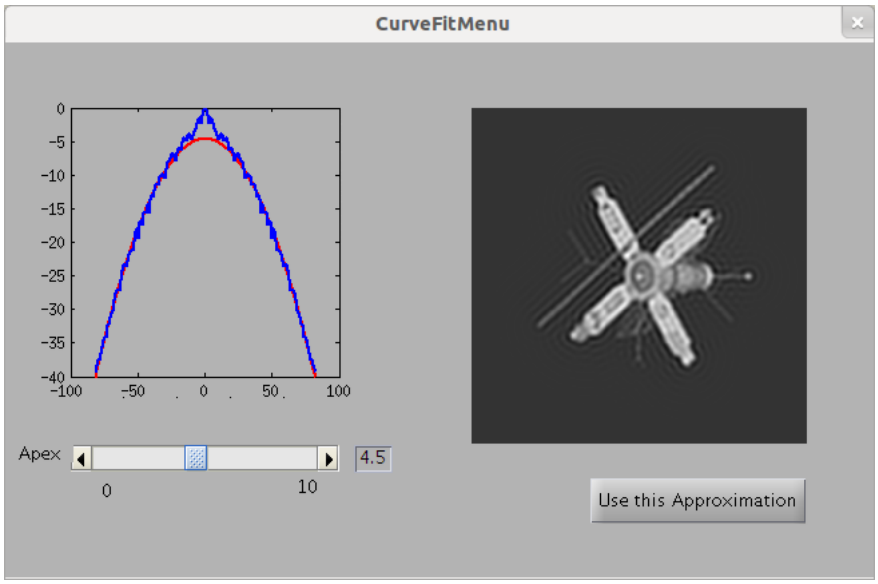
A.2. GUI for the APEX Method



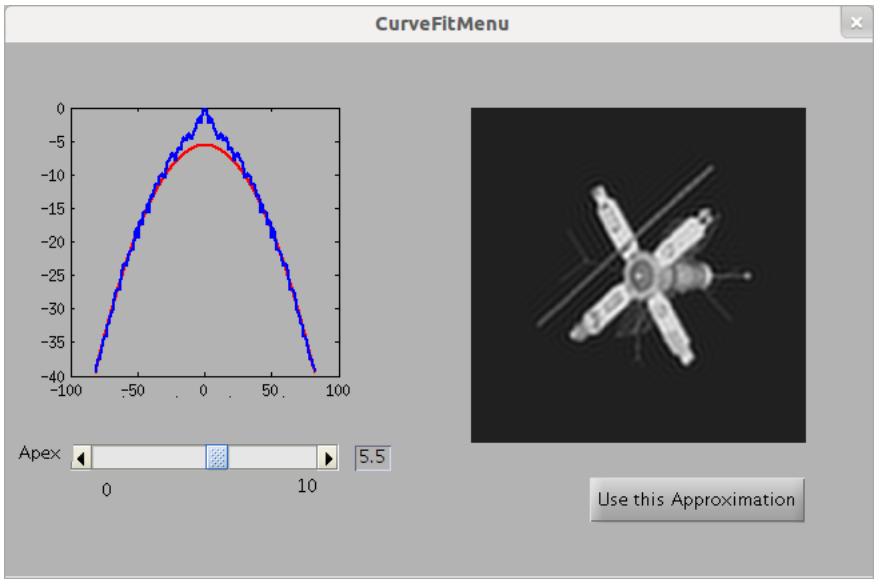
$\mathscr{A} = 2.5$



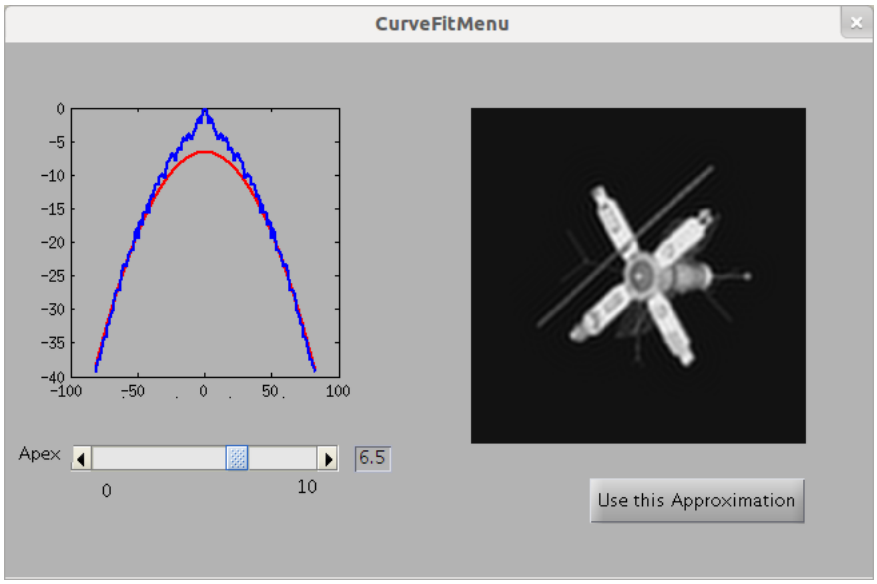
$\mathscr{A} = 3.5$



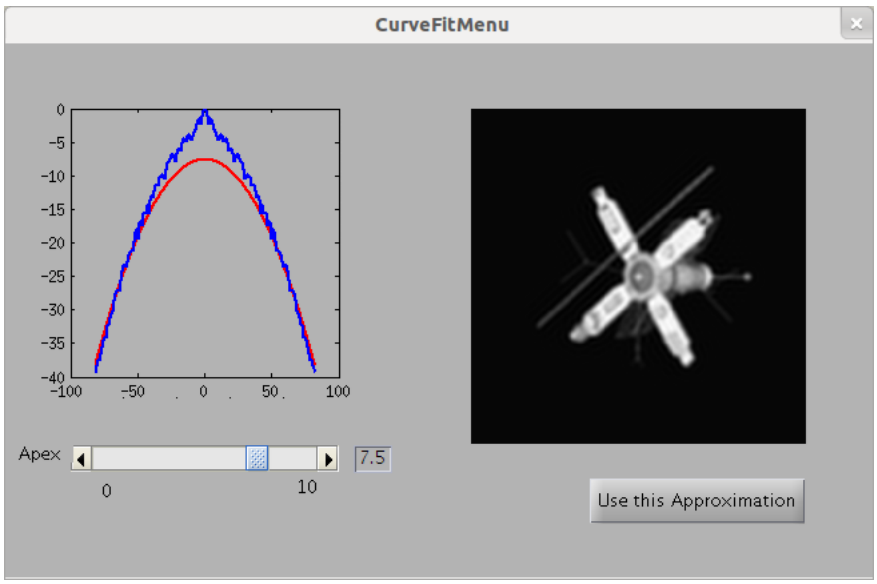
$\mathscr{A} = 4.5$



$\mathscr{A} = 5.5$



$\mathscr{A} = 6.5$



$\mathscr{A} = 7.5$

Bibliography

[1] H.M. Adorf. Restoring HST faint object camera images. In *European Southern Observatory Conference and Workshop Proceedings*, volume 38, pages 151–160, 1991.

[2] M. Bertero and P. Boccacci. Image restoration methods for the large binocular telescope (LBT). *Astronomy and Astrophysics Supplement Series*, 147(2):323–333, 2000.

[3] D.S.C. Biggs. *Accelerated Iterative Blind Deconvolution.* PhD thesis, University of Auckland, 1998.

[4] D. Bindel, J. Demmel, W. Kahan, and O. Marques. On computing Givens rotations reliably and efficiently. *ACM Transactions on Mathematical Software (TOMS)*, 28(2):206–238, 2002.

[5] M.B. Cannell, A. McMorland, and C. Soeller. Image enhancement by deconvolution. In J.P. Pawley, editor, *Handbook of Biological Confocal Microscopy*, chapter 25, pages 488–500. Springer, 2006.

[6] A.S. Carasso. Direct blind deconvolution. *SIAM Journal on Applied Mathematics*, 61(6):1980–2007, 2001.

[7] R.H. Chan and X.Q. Jin. *An Introduction to Iterative Toeplitz Solvers.* Fundamentals of Algorithms. SIAM, Society for Industrial and Applied Mathematics, 2007.

[8] J. Chung, S. Knepper, and J.G. Nagy. Large-scale inverse problems in imaging. In O. Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, chapter 2, pages 43–86. Springer, 2010.

[9] G. Cristóbal, P. Schelkens, and H. Thienpont. *Optical and Digital Image Processing: Fundamentals and Applications.* Wiley, 2011.

[10] P.J. Davis. *Circulant Matrices.* Chelsea Publishing Community, 1994.

[11] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Mathematics and Its Applications. Springer, 1996.

[12] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[13] G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[14] R.C. González and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2008.

[15] M. Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Vieweg + Teubner, 2006.

[16] P.C. Hansen, J.G. Nagy, and D.P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. Fundamentals of Algorithms. SIAM, Society for Industrial and Applied Mathematics, 2006.

[17] T.J. Holmes, D.S.C. Biggs, and A. Abu-Tarif. Blind deconvolution. In J.P. Pawley, editor, *Handbook of Biological Confocal Microscopy*, chapter 24, pages 468–487. Springer, 2006.

[18] D.A. Hope and S.M. Jefferies. Compact multi-frame blind deconvolution. *Optics Letters*, 36(6):867–869, 2011.

[19] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[20] S. Inoué. Foundations of confocal scanned imaging in light microscopy. In J.P. Pawley, editor, *Handbook of Biological Confocal Microscopy*, chapter 1, pages 1–19. Springer, 2006.

[21] M. Jiang, G. Wang, M.W. Skinner, J.T. Rubinstein, and M.W. Vannier. Blind deblurring of spiral CT images. *IEEE Transactions on Medical Imaging*, 22(7): 837–845, 2003.

[22] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, 1996.

[23] A. Mertins. *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms, and Applications*. Wiley, 1999.

[24] M.K. Ng. *Iterative Methods for Toeplitz Systems*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2004.

[25] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2006.

[26] P. Pankajakshan. *Blind Deconvolution for Confocal Laser Scanning Microscopy*. PhD thesis, University of Wisconsin, 2010.

[27] P. Pankajakshan, B. Zhang, L. Blanc-Féraud, Z. Kam, J.C. Olivo-Marin, and J. Zerubia. Blind deconvolution for diffraction-limited fluorescence microscopy. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 740–743. IEEE, 2008.

[28] JS Ploem and HJ Tanke. *Introduction to Fluorescence Microscopy*. Oxford University Press, 1987.

[29] V. Prasad, D. Semwogerere, and E.R. Weeks. Confocal microscopy of colloids. *Journal of Physics: Condensed Matter*, 19:113102–113126, 2007.

[30] T.J. Schulz. Multiframe blind deconvolution of astronomical images. *Journal of the Optical Society of America*, 10(5):1064–1073, 1993.

[31] P.J. Shaw. Comparison of widefield/deconvolution and confocal microscopy for three-dimensional imaging. In J.P. Pawley, editor, *Handbook of Biological Confocal Microscopy*, chapter 23, pages 453–467. Springer, 2006.

[32] G.W. Stewart. *Matrix Algorithms*. SIAM, Society for Industrial and Applied Mathematics, 1998.

[33] C.R. Vogel. *Computational Methods for Inverse Problems*. Frontiers in Applied Mathematics. SIAM, Society for Industrial and Applied Mathematics, 2002.

[34] M. Wax and T. Kailath. Efficient inversion of Toeplitz-block Toeplitz matrix. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(5):1218–1221, 1983.

[35] A.R. Webb. *Introduction to Biomedical Imaging*. IEEE Press Series in Biomedical Engineering. Wiley, 2003.