Blind Image Deconvolution Using The Sylvester Matrix

by

Nora Abdulla Alkhaldi

A thesis submitted to the
Department of Computer Science
in conformity with the requirements for
the degree of PhD

Sheffield University
England
November 2014

Abstract

Blind image deconvolution refers to the process of determining both an exact image and the blurring function from its inexact image. This thesis presents a solution of the blind image deconvolution problem using polynomial computations. The proposed solution does not require prior knowledge of the blurring function or noise level. Blind image deconvolution is needed in many applications, such as astronomy, remote sensing and medical X-ray, where noise is present in the exact image and blurring function. It is shown that the Sylvester resultant matrix enables the blurring function to be calculated using approximate greatest common divisor computations, rather than greatest common divisor computations. A developed method for the computation of an approximate greatest common divisor of two inexact univariate polynomials is employed here, to identify arbitrary forms of the blurring function. The deblurred image is then calculated by deconvolving the computed blurring function from the degraded image, using polynomial division. Moreover, high performance computing is considered to speed up the calculation performed in the spatial domain. The effectiveness of the proposed solution is demonstrated by experimental results for the deblurred image and the blurring function, and the results are compared with the state-of-the-art image deblurring algorithm.

I would like to dedicate this thesis to my teachers, family, and friends.

Acknowledgements

" لئن شكرتم لازيدنكم "

I would like to acknowledge and thank the king Abdullah scholarship program for allowing me to further my studies.

Many special thanks and appreciation go to my advisor Dr. Joab Winkler for his supervision, great efforts and support through the time of my PhD until the completion of this work.

This work would not have been carried out without the unconditional support and love given by my parents Abdullah and Dana: I am honoured and grateful to you for believing in me. My heartfelt gratitude extends to my brothers and friends for their assistance and help.

Also, I would like to thank my internal and external examiners Dr. Eleni Vasilaki and Professor Mahesan Niranjan for their comments and feedback. The final work has benefited greatly from their input.

Finally, "above every man of knowledge there is someone more knowledgeable"; to those who gave me assistance and helped to bring this work in this way, thank you all.

Contents

List of symbols

Mathematical variables

$\mathcal{F}$	Exact image
$\mathcal{G}$	Blurred image
$\mathcal{P}$	Point Spread function array
$\mathcal{E}$	Measurement error
$\mathcal{N}$	Additive noise
$F, G, P, \ldots$	Matrix representation of $\mathcal{F}, \mathcal{G}, \mathcal{P}, \ldots$ etc
$\mathbf{f}, \mathbf{g}, \mathbf{e}, \ldots$	vector representation of $F, G, E, \ldots$ etc
f_1, f_2	exact polynomials
g_1, g_2	inexact polynomials
$\bar{g}_1, \bar{g}_2$	preprocessed polynomials
$\tilde{g}_1, \tilde{g}_2$	corrected polynomials
$\hat{u}, \hat{v}$	exact polynomials
u, v	inexact polynomials
$S(f_1, f_2)$	Sylvester resultant matrix for the power basis polynomials $f_1(x)$ and $f_2(x)$
$S_k(f_1, f_2)$	Sylvester subresultant matrix of order k for the power basis $f_1(x)$ and $f_2(x)$

Mathematical operators

$\|\cdot\|$	$\|\cdot\|_2$ norm 2
$\otimes$	Convolution
$\dagger$	Pseudo inverse
$\oslash$	Element-wise division

Abbreviations

BID	Blind image deconvolution
PSF	Point spread function
SAR	Synthetic aperture radar
HST	Hubble space telescope
CT	Computerised tomography
MRI	Magnetic resonance imaging
RMSE	Root mean square error
SNR	Signal-to-noise ratio
NAE	Normalised absolute error
LSI	Linear spatially invariant
DFT	Discrete Fourier transform
BCCB	Block circulant with circulant blocks
BTTB	Block Toeplitz with Toeplitz blocks
BHHB	Block Hankel with Hankel blocks
BTHB	Block Toeplitz with Hankel blocks
BHTB	Block Hankel with Toeplitz blocks
TSVD	Truncated singular value decomposition
AR	Autoregressive modelling
GCD	Greatest common divisor
AGCD	Approximate greatest common divisor
SVD	Singular value decomposition
QR	QR decomposition
LS	Least squares
STLN	Structured total least norm
SNTLN	Structured non-linear total least norm
LP	Linear programm
DFT	Discrete Fourier transform
FFT	Fast Fourier transform

List of Tables

List of Figures

Chapter 1

Introduction

"A picture is worth a thousand words".

— Napoleon Bonaparte

Imaging is present in all aspects of our lives, and in research and development. A digital image is a numerical representation of an object or scene that is typically captured, stored, or modified using optical devices. However, the optical devices are not perfect, and various conditions can induce distortion in the recorded object and degrade the image.

There are many sources of blur, including imperfections in the lens, atmospheric turbulence, and the motion of the camera and/or object, and they result in the intensity of a given image not being recorded exactly, such that the intensity of a given pixel in the observed image is affected by its neighbouring pixels in the exact image [38]. Moreover, the recorded image is commonly contaminated with random noise. This noise can be generated from the optical device itself, in particular an image sensor, or from digitisation and quantisation systems, and it

can lead to measurement errors [54].

The light intensity in each pixel of the blurred image is spread by the convolution operation [8, 38]. The blurring function is often referred to as a *point spread function* (PSF) and it describes the shape and the pattern of the blurring function. In some applications the PSF is spatially variant and in others it is spatially invariant. The spatially variant PSFs vary over the blurred image such that each pixel contains a different PSF. Since it is difficult to estimate how the blur will behave over the image, it is often assumed that the PSF is spatially invariant [38].

In *classical image restoration* the exact image is reconstructed from its blurred version by performing a backward operation to the image construction, basically by removing the blur which assumed to be known exactly [54]. Several applications need image restoration, including video-conferencing, astronomical and medical imaging, remote sensing, barcode readers and synthetic aperture radar (SAR). However, it may be difficult or impossible to know the blur information *a priori*, and there may only be partial knowledge of the true image given [33, 54, 81]. This makes it a challenging problem and therefore the term *blind image deconvolution* (BID) has been introduced [38, 51, 57]. BID refers to the process of identifying the true image from its blurred version, possibly using prior information of the PSF and the noise level [51, 54].

Clearly, classical image restoration methods are not practical for many applications in image processing, where the PSF and noise information are not provided. The solutions of the BID problem, therefore, employ additional information to restore a true image, such as physical properties of a given degraded image including image dimensions, resolution, or statistical properties such as entropy

(which represents the randomness in an image).

This observation has helped many researchers to reduce the various effects of degradation, both blur and noise, and to estimate an approximation of the true image. These solutions can be implemented in many fields that require image restoration. The following section provides some examples of the application of the BID problem.

1.1 Applications of blind image deconvolution

This section considers some of the main applications that uses the solutions of the BID problem to reconstruct a true image.

1.1.1 Astronomical imaging

Astronomy is a field that has contributed to the evolution of image restoration. An astronomical image is an essential source of information for astronomers to study cosmological objects, which are millions of light-years distant from Earth. These objects are faint and require astronomers to use larger telescope apertures; therefore curved mirrors are used instead of lenses to make observations. Images are normally obtained using special ground-based telescopes, such as the one in the Andes in Chile, and the two in Mauna Kea in Hawaii. However, the distortion caused by turbulence in the atmosphere prevents a telescope from capturing a clear image of an object. In particular, the light transmitted from stars is sent through space undistorted until it hits the atmosphere, where it becomes distorted [8, 79].

To reduce the atmospheric fallout on the recorded images high quality space-based telescopes are used, such as the Hubble space telescope (HST). However, this is an expensive solution. A simple diagram (Figure 1.1) shows how astronomical imaging is obtained using ground- and space-based telescopes. The main improvement in the manufacture of telescopes is the correction of spherical aberration in mirror.

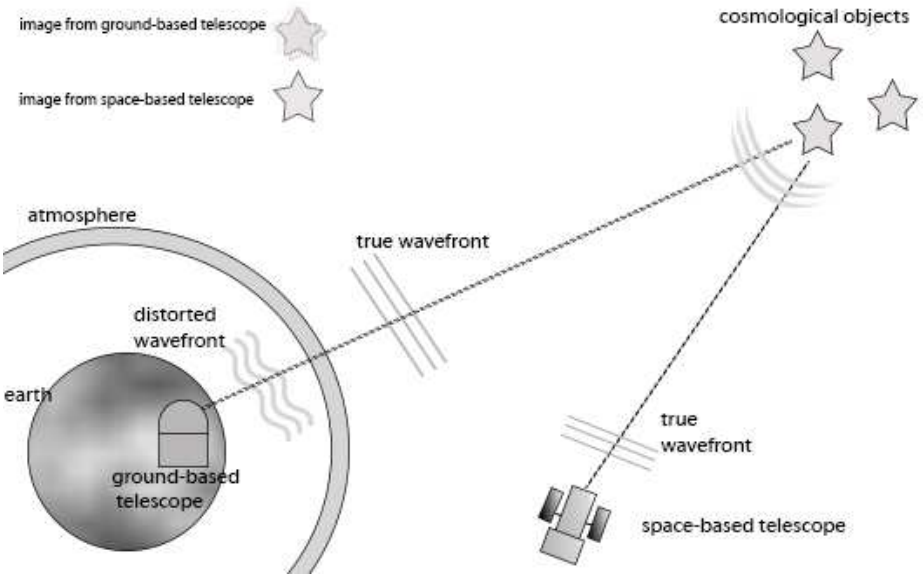


Figure 1.1: The light emitted from stars is distorted by turbulent atmosphere. Observations using ground-based telescopes include distortion effects that have to be reduced by using BID methods; the alternative is to employ space-based telescopes, [8].

Alternatively, BID methods are employed to reduce the atmospheric effects in the observed image by estimating the distortion function or the PSF. Figure 1.2 provides an example of an image of cosmological objects taken by HST and two blurred versions of it. Deconvolution methods in astronomy have been inves-

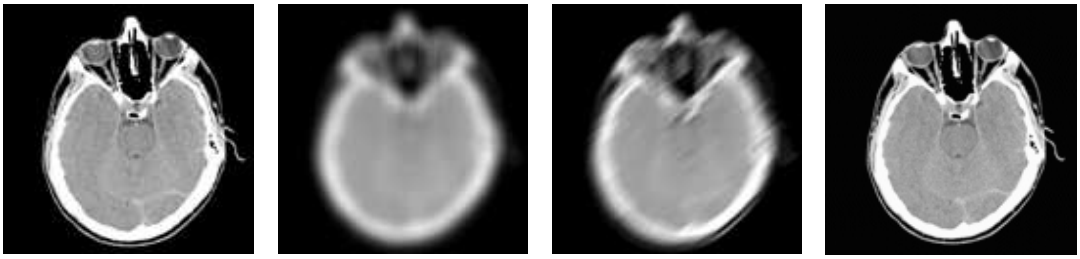
(a) **(b)** **(c)** **(d)**

Figure 1.2: *Astronomical image from HST.* (b) and (c) are two distorted images, that are generated using MATLAB function `conv2`, of an original image in (a); (d) is a restored image using the method described in this thesis.

tigated by many researchers [22, 82] and an overview is presented in [79].

1.1.2 Medical imaging

Image restoration techniques have been used in modern medicine as diagnostic and therapeutic procedures [8]. The objective of medical imaging research is to remove noise from distorted images without losing important information. For example, identifying a tumour in a noisy background image is a challenging problem. Medical images can be observed using many technologies such as X-ray, computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound. Figure 1.3 provides an example of image degradation using one of the medical technologies. This type of problem has encouraged many scientists to study the mathematical and statistical properties of medical images which often are noise contaminated [34, 73, 74].

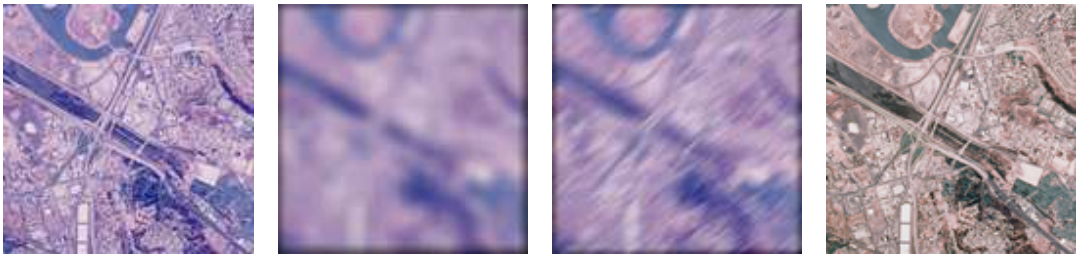


(a) **(b)** **(c)** **(d)**

Figure 1.3: *A medical CT imaging example.* (b) and (c) are two distorted images of an original image in (a) obtained using MATLAB function `conv2`, and contaminated with different levels of noise; (d) is a restored image using the methods described in this thesis.

1.1.3 Remote sensing

Remote sensing refers to the technology of collecting images and information of a particular phenomenon that occurs on earth or outside the atmosphere, from a distance [15]. Remote sensors can be attached to several platforms, such as satellites and unmanned airborne vehicles, so that it becomes easy to cover wider regions. The instruments, such as cameras, work by observing the electromagnetic radiation reflected from an object [15]. However, an image obtained using a satellite system passes through various layers of Earth's atmosphere, resulting in a degraded image, and degradation also occurs if camera motion is introduced. Although the development of remote sensing technology is growing rapidly, the demand of BID remains [15, 90]. The importance of the BID algorithms in remote sensing can be seen in Figure 1.4.



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

Figure 1.4: *An aerial map image of San Diego.* (b) and (c) are two distorted images of an original image in (a), using MATLAB function `conv2`, and contaminated with noise; (d) is a restored image using the methods described in this thesis.

From the previous discussion, it can be seen that the BID becomes an essential procedure in several applications. This thesis is devoted to solve the BID problem, and the next section outlines the description of this research.

1.2 Description of the thesis

In the next sections the aims, contributions and structure of the thesis will be discussed.

1.2.1 Scope of the research

The objective of this research is to solve the BID problem, such that prior knowledge of the exact image and the PSF are not required. The proposed BID method is basically divided into two separate stages: the first aims to estimate the PSF

and the second to restore the image. The difficulty in performing this algorithm comes from its sensitivity to additive noise that arises from different sources, and which makes the BID an ill-posed problem. In particular, an ill-posed problem occurs when a continuous change in the input produces a discontinuous change in the output information.

The PSF can be obtained using a greatest common divisor (GCD) computation of two blurred versions. However, introducing additive noise to the problem reduces to an approximate greatest common divisor (AGCD) computation. These calculations are carried out using the Sylvester resultant matrix approach, such that two AGCD computations are performed.

Although the AGCD computations have been employed for image deblurring previously [2, 60, 69], this work differs from other works, in that that prior knowledge of the PSF and the noise level are not required to restore an image from its blurred version/versions.

The next section discusses the main contributions of the thesis.

1.2.2 Contributions of the thesis

Figures 1.2, 1.3, and 1.4 illustrate that the problem of image blurring exists in several applications during image acquisition. The proposed image restoration algorithm in this thesis can solve the following situations:

- If one distorted image is given, then a fast deconvolution method is applied to restore the separable PSF and the image. This involves two AGCD computations on two distinct rows and on two distinct columns of the distorted

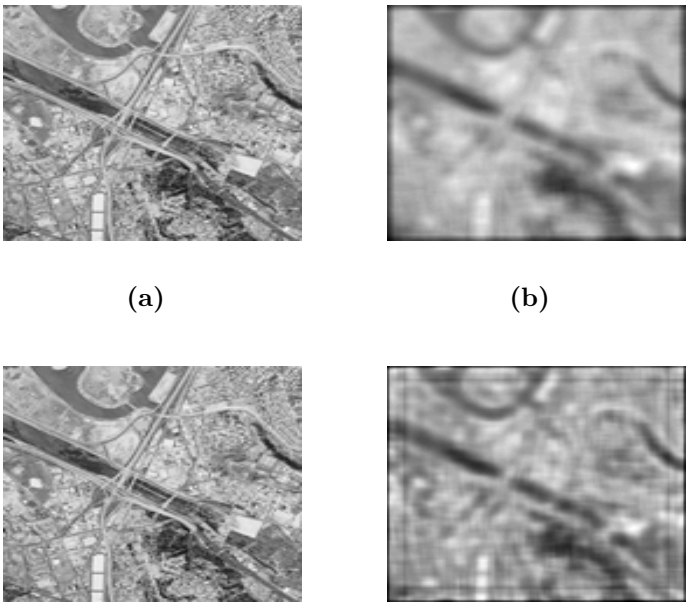
image. Then the deconvolution step is performed.

- If two distinct distorted images are given, such that each consists of similar blur effects (e.g. taken by the same camera), then a fast deconvolution method is applied to restore the separable PSF and the image. The PSF is obtained by performing two AGCD computations; one is performed on two distinct rows from each given image and the other is performed on two distinct columns from each image. This is followed by a deconvolution procedure for each image.

- If two distinct distorted images are given, such that each consists of similar blur effects, and if the PSF is not separable, then a high performance parallel computation is performed to restore the PSF and the image. To be more precise, if the PSF cannot be decomposed into a horizontal and vertical components, then it said to be nonseparable. The PSF is obtained by performing AGCD computations; one by considering every row from each given image and the other by considering every column from each image. Then, a deconvolution procedure is performed for each image using the computed PSF.

The proposed work can be operated on SNR lower than 30 dB, while some previous works operate on much higher SNR (e.g. 45 dB in [69] and 50 dB in [24]).

The following example shows some of the results obtained by applying the proposed BID algorithm. These results are compared with state-of-the-art image restoration methods. More results, performance analysis, and comparisons are included in later in Chapter 8.



(a) **(b)**

(c) **(d)**

Figure 1.5: *Image restoration results.* (a) an original aerial image; (b) a distorted version of (a); (c) and (d) are the restored images using the implemented approach in this thesis and the Richardson-Lucy method, respectively.

Example 1.1. Figure 1.5, shows the image restoration output of a blurred aerial image given in (b). The performance of image restoration methods is measured using different quantities such as root mean squares error (RMSE), normalised absolute error (NAE), and with signal-to-noise ratio (SNR).

RMSE and NAE measure the differences between the blurred and the restored images, and the SNR is defined as

$$\mathrm{SNR}_{dB} = 10 * \log_{10} \frac{P_{signal}}{P_{noise}}, \tag{1.1}$$

where P stands for average power (i.e. mean square error for ground-truth signal and background noise). The measurement techniques are described in detail in Chapter 8.

Table 1.1: *Comparison of* 5 *deblurring methods.* M1: The method described in this thesis, M2: Lucy-Richardson, M3:Wiener filter, M4: Regularised filter, M5: Maximum likelihood.

Image	*Deblurring methods*	RMSE	NAE	SNR
Aerial	M1	$4.98108e-06$	$4.44995e-04$	$5.95483e+01$
	M2	$8.45910e-04$	$1.28542e-01$	$3.19328e+00$
	M3	$2.32220e-05$	$8.13516e-04$	$4.79120e+01$
	M4	$5.11691e-04$	$8.60430e-02$	$9.42049e+00$
	M5	$8.43885e-04$	$1.28195e-01$	$3.20511e+00$
Grass	M1	$1.09234e-05$	$1.66468e-03$	$4.01537e+01$
	M2	$1.05321e-03$	$1.56484e-01$	$4.81140e-01$
	M3	$1.70980e-05$	$2.68587e-03$	$3.62553e+01$
	M4	$5.66471e-04$	$8.85061e-02$	$5.80275e+00$
	M5	$1.05514e-03$	$1.56837e-01$	$4.84129e-01$

The restored image using the proposed BID method shows an improvement, as can be easily seen in Figure 1.5 (c). The SNR has increased significantly from 1.9441 dB to 59.5483 dB; more importantly, this image is restored with no prior knowledge of the PSF and the noise level. The result is compared with a number of built-in MATLAB functions including the Richardson-Lucy method in Figure 1.5 (d); however, most of these functions require the PSF to be known *a priori*. Similarly, Figure 1.6 compares the implemented BID method with the Richardson-Lucy deconvolution method. The SNR has improved remarkably from -0.9363 dB to 40.1537 dB using the implemented BID.

Table 1.1 provides a comparison of 5 deblurring methods. Most of the existing methods require an *a priori* knowledge of the PSF and the additive noise, but the proposed method of BID gives a much better result with no prior assumption of the PSF and noise.

The outline of the thesis is presented in the following section.

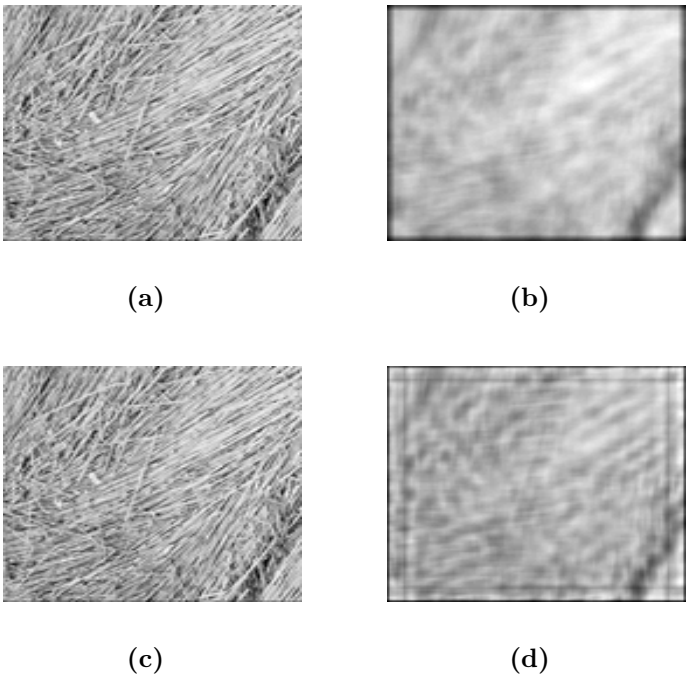


Figure 1.6: Image restoration results, (a) an original grass image, (b) a distorted version of (a), (c) and (d) are the restored images based on the implemented approach and Richardson-Lucy methods, respectively.

1.2.3 Overview of the thesis

The rest of this thesis is divided into nine chapters.

Chapter 2 introduces some fundamental terms needed in image formation, including image convolution. This chapter also presents an overview of some of the most frequently used approaches in image deconvolution. Chapter 3 provides an overview of the GCD and an AGCD definitions, presents various methods for computing an AGCD, and discusses the adopted definition of an AGCD and its computation-based method. Chapter 4 outlines the proposed method for image deblurring; it first defines the problem in polynomial form and then describes a solution for the BID problem in one-dimensional and two-dimensional cases, including separable and nonseparable PSF. This solution is carried out using Sylvester resultant matrix and subresultant matrices of two polynomials, as described in Chapter 5. It discusses the Sylvester resultant matrix properties, how it relates to the rank estimation problem, and the preprocessing operations required to adjust the variation of magnitude among the coefficients of the Sylvester matrix. Chapter 6 highlights the methods used in this research to compute the PSF, i.e. using an AGCD of two inexact polynomials. This includes two techniques for computing the degree of an AGCD namely: the first principle angle and residual methods. The computation method of the coefficients of an AGCD is described then, using a structured low rank approximation. Chapter 7 presents two methods for the PSF identification. The first part considers the restoration of separable PSF in two cases: single degraded image, and two degraded images. The second section describes the identification method of arbitrary PSF, including separable and nonseparable PSFs. The deconvolution of the computed PSF

from the degraded image is discussed in Chapter 8. This chapter also discusses some computational specifications of the implemented algorithm. The experimental results of the designed image restoration using polynomial computations is shown in Chapter 9. A summary of the described method and its results are presented in chapter 10, along with suggestions for future work.

1.3 Summary

A summary of the image restoration problem and some of its applications have been introduced in this section, along with an overview of the main aim and contribution of this thesis. The final section provided an outline of the thesis.
It is important to study and analyse the behaviour of image blurring, including image distortion model, the properties of the PSF, and the convolution mechanism, in order to restore an image. In the next chapter, the image formation problem will be presented in greater detail, along with a literature review on image restoration methods.

Chapter 2

Image restoration

2.1 Introduction

It was shown in Chapter 1, image restoration is an essential procedure in many applications, thus this chapter is devoted to providing an overview of image restoration. A blurred image is obtained by convolution of an exact image and distortion effects. Section 2.2 discusses the degradation model of an image. Blur characterisation and properties are introduced in Section 2.3, as well as some of its forms including motion blur, out-of-focus blur, two-dimensional Gaussian blur, and the Moffat function. As a result of introducing a blur to an exact image during image acquisition, a boundary conditions problem arises; this will be discussed in Section 2.4. Section 2.5 considers the convolution process in one-dimensional (1D) and two-dimensional (2D) problems. The final section gives an overview of previous works in the image restoration area.

2.2 Image distortion model

To design a mathematical model of a blurred image, it is necessary to represent the image as an array of numbers. An image of two or three dimensions consists of arrays of elements called pixels. Each pixel is assigned an intensity between 0 and 256 units, on a gray scale or colour scale. The RGB image stores three components per pixel, or three channels, that correspond to red, green and blue scales respectively. For example, the intensity values $(1,0,0),(0,1,0)$, and $(0,0,1)$ are referred to as a pure red, green and blue images respectively [38]. The following discussion will be restricted to one channel or grey-scale images, since the same process can be applied on each channel separately for RGB images.
The operation of the imperfection in blurred images caused by blur is referred to as a convolution. In mathematics the *convolution* is an operation that averages the amount of overlap between two functions. It occurs in both continuous and discrete form, and can be used in many applications for mathematics and engineering [13].

Definition 2.1. Let $f(i,j)$ and $p(i,j)$ be two functions. Then the convolution of p and f in discrete form is a function g that is given by

$$\begin{aligned} g(i,j) &= (p \otimes f)(i,j), \\ &= \sum_i \sum_j p(i-k, j-l) f(i,j) \end{aligned}$$

where $\otimes$ stands for convolution. The output function $g(i,j)$ is a weighted average of $f(i,j)$ and $p(i,j)$.

The transformation from input function into output function is assumed to be

linear; moreover, it is assumed to be shift invariant [67]. In the image processing context, each pixel value of the blurred image is equal to the weighted average of the corresponding pixel value in the exact image and its neighbouring pixels, such that weights are given by the PSF [38]. This means that in the image restoration problem the assumptions of linearity and shift invariant (LSI) properties hold.

The most important point behind acquiring an acceptable reconstructed image is to design an appropriate a model of the distortion. But it is difficult to assume such a model because of the lack of information of the true image and the blur; only robust image restoration methods can challenge this limitation [54]. A mathematical model, however, that considers an observed image $\mathcal{G}$ as an altered version of an exact image $\mathcal{F}$ can be modelled as

$$\mathcal{G} = \mathcal{F} \otimes \mathcal{P} + \mathcal{N}, \tag{2.1}$$

where $\mathcal{P}$ and $\mathcal{N}$ describe the PSF and additive noise in the blurred image, respectively. This model might not be an exact representation of the problem, but it gives an acceptable solution in many image restoration problems [54]. The random noise $\mathcal{N}$ in image $\mathcal{G}$ could be generated from different sources, including environment conditions and lens deficiency. The model in Eq. 2.1 is defined in spatial domain as below

$$g(i,j) = \sum_{i}^{M-1} \sum_{j}^{N-1} p(i-k, j-l) f(i,j) + n(i,j).$$

Suppose F, G, P, and N are the matrices that identify the exact image $\mathcal{F}$, the blurred image $\mathcal{G}$, the PSF array, and the additive noise $\mathcal{N}$ respectively, then

$$\mathbf{g} = \nu(G), \qquad \mathbf{f} = \nu(F), \qquad \mathbf{n} = \nu(N),$$

where ν denotes that $\mathbf{g}, \mathbf{f}$ and $\mathbf{n}$ are vectors obtained by stacking the columns of

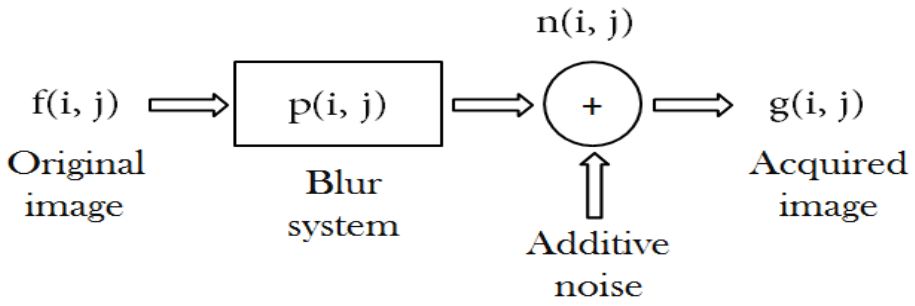


Figure 2.1: Image formation model.

the matrices G, F, and N respectively. It follows that the blurred image can be expressed in matrix-vector form as

$$\mathbf{g} = H\mathbf{f} + \mathbf{n}, \tag{2.2}$$

See also Figure 2.1. The matrix H is ill-conditioned [57], and contains the coefficients of the PSF with a dimension equal to the image size. The PSF reduces the quality of image $\mathcal{F}$, as does any additive noise. If no additive noise is introduced to the convolution problem, then $\mathbf{n} = 0$. The determination of the original image $\mathcal{F}$ requires the calculation of an approximation of the vector $\mathbf{f}$. The main aim in image restoration is to remove the blur from the input image, since it is considered to be more significant than the additive noise [57].

The next section defines the systematic introduced blur – the PSF and its properties.

2.3 Blurring function

Recall from Chapter 1 that the PSF describes the blurring function. More precisely, the PSF identifies the response that is the output of an imaging system to a point source that is the input. This section considers some different shapes of the

PSF caused by environmental conditions. The blurring in the image restoration holds several assumptions, as listed below.

1. The PSF is spatially invariant in many applications but not all. More precisely, it is independent in each pixel location. This implies the PSF is equally distributed along x and y axes in the blurred image $\mathcal{G}$ [38]. By contrast, the spatially variant PSF is computationally difficult to construct, since every pixel has a different PSF [6].

2. The light intensity of the PSF is confined to a small area around its centre, such that it is zero beyond a given distance from its centre; therefore, blurring is a local phenomenon [38].

3. The dimensions of P, which is the matrix form of PSF array, are much smaller than those of $F \in \mathbb{R}^{m \times n}$, such that P contains all information about the blurring of F [38].

4. The sum of pixel values of the PSF array is equal to one, since the imaging process captures all the light [38, 54]. Consider matrix $P \in \mathbb{R}^{c \times d}$; then
$$\sum_{k=0}^{c-1} \sum_{l=0}^{d-1} p_{k,l} = 1, \tag{2.3}$$
where k, l define the location of each pixel intensity of the PSF.

5. The coefficients of matrix P have only positive values, since the nature of image formation cannot be negative [54].

The PSF is, therefore, a function of distance between the influencing pixel in $\mathcal{F}$ and the influenced pixels in $\mathcal{G}$ [38]. However, if no blur is included in the observed image then the PSF is considered to be a unit pulse [54]. More precisely, the sum

of the entries of P is equal to one.

The pattern of the PSF is often influenced by the optical device that is used to record the blurred image or by other physical or environmental events. Typical examples of the PSF model are given below:

1. **Out-of-focus blur** [38, 54] is caused by the aperture and the focal length of the lens [54]. In particular, defocusing the lens at the acquisition process results in the loss of some important features of the image, such as sharpness. A system with a circular aperture can be expressed as a uniform disk with radius R. If the centre of the PSF array defined as the matrix P is given by (k, l) then the out-of-focus blur for each element $p_{i,j}$ is

$$p_{i,j} = \begin{cases} \frac{1}{\pi R^2} & \text{if} \quad \sqrt{(i-k)^2 + (j-l)^2} \leq R^2, \\ 0 & \text{otherwise.} \end{cases}$$

   Figure 2.2 (a) shows the effect of out-of-focus PSF on image acquisition. The shape of out-of-focus PSF can be seen in Figure 2.3, where (a) represents the 2D surface plot and (d) represents the 2D image of the PSF array.

2. **Motion blur** is present as a result of object movement or camera shake or both, during the exposure time; see Figure 2.2 (b). The 1D motion blur defines a proportional relation between the objects to be seen and the camera along the horizontal direction [54].

3. **The 2D Gaussian blur** [38] is given by

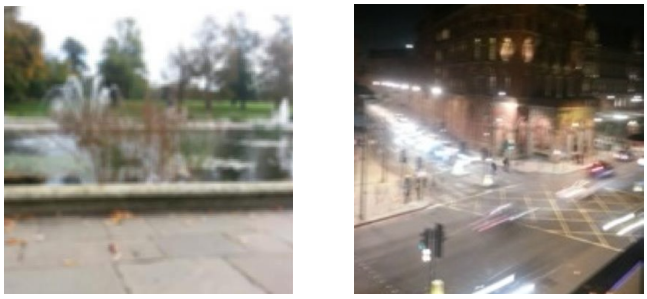
$$p_{i,j} = \exp\left(-\frac{1}{2}\begin{bmatrix} i-k & j-l \end{bmatrix} \begin{bmatrix} s_1^2 & \sigma^2 \\ \sigma^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix} \right). \tag{2.4}$$

Here the parameters s_1, s_2, and σ determine the width and orientation of the PSF, centred at point (k, l) in P. This type of blur has an exponential decay shape from the centre, typically truncated when its values are decayed sufficiently. The PSF is assumed to be horizontally and vertically symmetrical if $\sigma = 0$, and rotationally symmetric if $s_1 = s_2$. In this case the PSF is said to be separable, where it can be represented as a multiplication of a column-vector by a row-vector. Figure 2.3 illustrates the surface of 2D Gaussian blur in (b), and displays it as an image (c).

4. The **Moffat function** is a blur usually caused by astronomical devices, and is given by

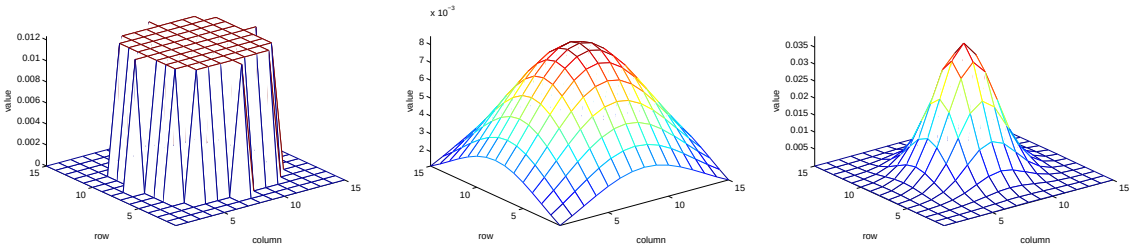
$$p_{i,j} = \exp\left(1 + \left(\frac{i-k}{s_1}\right)^2 + \left(\frac{j-l}{s_2}\right)^2\right)^{-\beta}, \tag{2.5}$$

in which the parameters s_1, s_2, and β control the width and the decay parameters of the PSF. The decay rate of a Moffat PSF is much slower than a Gaussian PSF. Figure 2.3 (c) shows the 2D surface of the Moffat PSF and as an image in (f) [38, 82]. It is clear that the Gaussian function is a limited condition of the Moffat function, such that $\beta \to \infty$ [82]. With regard to narrow PSFs, the Moffat function is numerically more accurate than the Gaussian function [82]. The Moffat functions are also rotationally symmetric if $s_1 = s_2$ [38] .

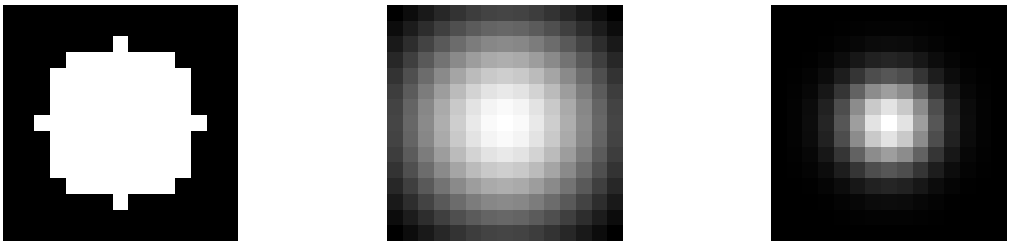


(a) **(b)**

Figure 2.2: *Different sources of image blurring.* (a) Out-of-focus blur; (b) Motion blur.



(a) Out-of-focus blur **(b)** 2D Gaussian blur **(c)** Moffat blur



(d) Out-of-focus blur **(e)** 2D Gaussian blur **(f)** Moffat blur

Figure 2.3: *Examples of three PSFs.*

As was pointed out previously, the sum of the elements of the PSF in P are scaled so that their sum is equal to one. The following example describes the pattern of

the PSF.

Example 2.1. Suppose a 2D PSF array is given by a non-zero matrix $P \in \mathbb{R}^{7\times 7}$, such that

$$\begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} & p_{04} & p_{05} & p_{06} \\ p_{10} & p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} \\ p_{20} & p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} \\ p_{30} & p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{36} \\ p_{40} & p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{46} \\ p_{50} & p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{56} \\ p_{60} & p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} \end{bmatrix}.$$

This PSF is rotationally symmetric, if

$$\begin{aligned} p_{00} &= p_{06} = p_{60} = p_{66}, \quad p_{01} = p_{05} = p_{10} = p_{16} = p_{50} = p_{56} = p_{61} = p_{65}, \\ p_{02} &= p_{04} = p_{20} = p_{26} = p_{40} = p_{46} = p_{62} = p_{64}, \quad p_{03} = p_{30} = p_{36} = p_{63}, \\ p_{11} &= p_{15} = p_{51} = p_{55}, \quad p_{12} = p_{14} = p_{25} = p_{21} = p_{41} = p_{45} = p_{52} = p_{55}, \\ p_{13} &= p_{31} = p_{35} = p_{53}, \quad p_{22} = p_{24} = p_{42} = p_{44}, \quad p_{23} = p_{32} = p_{34} = p_{43}, \end{aligned}$$

and

$$\sum_{i=0}^{6}\sum_{j=0}^{6} p_{ij} = 1.$$

The pixel values in matrix P can be written as the following:

$$\begin{aligned} p(x,y) &= p(6,6)x^6y^6 + p(6,5)x^6y^5 + p(6,4)x^6y^4 + \cdots + p(0,2)y^2 + p(0,1)y + p(0,0), \\ &= \sum_{i=0}^{6}\sum_{j=0}^{6} p(6-i,6-j)x^{6-i}y^{6-j}, \\ &= 1. \end{aligned}$$

If the PSF array is symmetrical along x and y directions, it said to be separable,

such that,

$$p(x,y) = \begin{bmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 \end{bmatrix} P \begin{bmatrix} 1 & y & y^2 & y^3 & y^4 & y^5 & y^6 \end{bmatrix}^T .$$

A blurry image is formed by the convolution between the PSF array and an exact image. The convolution procedure extends outside the border of an exact image, and therefore it is a crucial to consider the boundary conditions problem when performing the convolution. Section 2.4, discusses the boundary conditions problem.

2.4 Boundary conditions

This section previews the standard types of boundary conditions at the border of an exact image. Due to the digitisation process that occurs when capturing an image, the information in some pixels can overflow to the neighbouring pixels. The convolution process of the exact image and the PSF includes the out-of-border pixels, causing a blurred image [54]. Clearly some of this information at the border of the exact image will be lost; however, knowledge of the information is required in the inverse problem. Therefore, to mend this effect, it is important to predict the structure of the exact image outside the border. In particular, the boundary conditions can be incorporated in the matrix H introduced in Eq. 2.2. Failing to do so may result in oscillations at the border of a restored image with bad quality, as can be seen in Figure 2.4.

The boundary conditions problem makes the following assumptions: the zero boundary condition, the periodic boundary condition and the reflexive boundary condition.

(a) **(b)**

Figure 2.4: *The boundary conditions problem in image restoration.* (a) presents the blurred image using the linear motion filter of size 5×5 and angle $45°$. (b) is the restored image using a Matlab built-in function, showing oscillations at its border.

Zero boundary condition [38]. It is usually assumed for images with mostly black background, for instance in astronomical images. It creates a larger image, such that the original image is placed in the centre and the remaining pixels of the larger image are set to zero. However, this assumption is not appropriate with images that have a considerable variation between the inner and outer pixel values (with respect to the border). Figure 2.5 (b) shows an example of an image with zero boundary condition. The zero boundary condition implies that matrix H is constructed using a block Toeplitz with Toeplitz blocks matrix (BTTB) structure in the image convolution. The BTTB matrix can be written as

$$H = \begin{bmatrix} T_0 & T_{-1} & \cdots & T_{1-m} \\ T_1 & T_0 & \cdots & T_{2-m} \\ \vdots & \vdots & \ddots & \vdots \\ T_{m-1} & T_{m-2} & \cdots & T_0 \end{bmatrix}, \qquad i = 0, \cdots, m-1, \tag{2.6}$$

where each T_i represents a Toeplitz structure.

Periodic boundary condition . This assumption, commonly used in image processing, assumes a block circulant with circulant blocks matrix representation (BCCB) [2, 38]. A matrix of BCCB is formed by blocks of circulant matrices, where each column is repeated continuously with a shift in position. The BCCB matrix can be written as

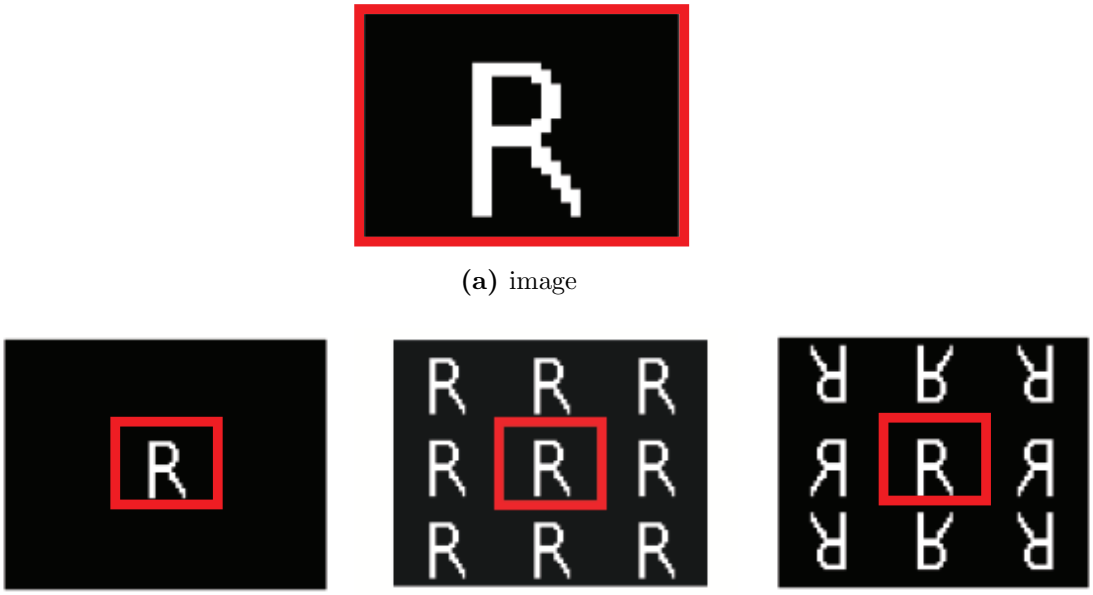
$$H = \begin{bmatrix} C_0 & C_{m-1} & \cdots & C_1 \\ C_1 & C_0 & \cdots & C_2 \\ C_2 & C_1 & \cdots & C_3 \\ \vdots & \vdots & \ddots & \vdots \\ C_{m-1} & C_{m-2} & \cdots & C_0 \end{bmatrix}, \qquad i = 0, \cdots, m-1, \tag{2.7}$$

where each C_i is a circulant matrix [68].

The structure of the periodic boundary condition is based on embedding the exact image repeatedly in a larger image. Figure 2.5 (c) manifests this type of boundary condition. The periodic boundary condition is mainly implemented using discrete Fourier transform (DFT) computations [68].

Reflexive boundary condition . This assumes that the pixel values outside the border of an exact image are a reflection of those inside, as shown in Figure 2.5 (d). The structure of the reflexive boundary condition is based on a mixture of several block-structured matrices including, $BTTB$ and other

- $(BHHB)$: Block Hankel with Hankel blocks.
- $(BTHB)$: Block Toeplitz with Hankel blocks.
- $(BHTB)$: Block Hankel with Toeplitz blocks. The Hankel matrix is a



(a) image **(b)** zero **(c)** periodic **(d)** reflex

Figure 2.5: *The boundary conditions assumptions.* (a) is an exact image; (b), (c) and (d) show the effects of the zero boundary condition, the periodic boundary condition and the reflexive boundary condition, respectively, where the red lines shows the border of (a).

square matrix which can be expressed as

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{m-1} \\ h_1 & h_2 & h_3 & \cdots & h_m \\ h_2 & h_3 & h_4 & \cdots & h_{m+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m-1} & h_m & h_{m+1} & \cdots & h_{2m-2} \end{bmatrix}, \qquad i = 0, \cdots, m-1, \quad (2.8)$$

where each skew-diagonal is constant. More details of these matrices are given in [38, 40, 52].

The solution of the boundary condition problem varies depending on the used domain – in particular, if the spatial domain or Fourier domain is implemented [54]. The spatial domain can uses zero, periodic, and reflexive boundary conditions,

while Fourier domain assumes the periodic boundary condition. The missing information of pixels located outside the degraded image are needed to solve the deconvolution problem. The implementation of any condition requires to build a large structured matrix, which is computationally expensive process. In spatial domain, it more appropriate to consider the separable PSFs, since it can speed up the computation using their decomposition property [61]. However, Fourier domain assumes the periodic boundary condition, which can solve the deconvolution problem faster using DFT. The next section describes the mathematical behaviour of the convolution problem in 1D and 2D.

2.5 Convolution

The convolution of two functions was introduced in Definition 2.1. To solve the BID problem, it is important to define the characterisation of the blurring matrix H in Eq. 2.2. As discussed in Section 2.4, the structure of H is dependent on the imposed boundary condition. Section 2.5.1 describes how the convolution occurs in the image blurring. Section 2.5.2 explains the convolution using separability of a PSF.

2.5.1 Two-dimensional convolution problem

This section considers the convolution matrix in 2D form. The blurred image $\mathcal{G}$ is obtained as a result of the convolution of an exact image $\mathcal{F}$ and a 2D PSF array. The pixel values of the blurred image $\mathcal{G}$ are stored in the matrix G. This problem is modelled in matrix-vector form as in Eq. 2.2, where the images $\mathcal{F}$ and $\mathcal{G}$ are transformed into column vectors $\mathbf{f}$ and $\mathbf{g}$ respectively, and $\mathbf{n}$ is the noise

column vector. Since the information of the PSF is influenced by the introduced boundary condition at the border of $\mathcal{F}$, a large matrix $H \in \mathbb{R}^{mn\times mn}$ is established, such that H contains the information of matrix $P \in \mathbb{R}^{m\times n}$ of the PSF array. The structure of the blurring matrix H is determined by the introduced boundary conditions explained in Section 2.4. The 2D convolution process, when $\mathbf{n} = 0$ in Eq. 2.2, can be illustrated by the next example.

Example 2.2. [38]. Consider an exact image $\mathcal{F}$, a PSF array, and blurred image $\mathcal{G}$ in the noise free environment, each of size 3×3, such that their matrices representation F, G and $P \in \mathbb{R}^{3\times 3}$ are

$$F = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \qquad P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}, \qquad G = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}.$$

First, the PSF matrix P must be rotated by 180° degrees. This is accomplished by flipping its rows, resulting in matrix P_1, and then flipping its columns, resulting in matrix P_2, as below.

$$P_1 = \begin{bmatrix} p_{31} & p_{32} & p_{33} \\ p_{21} & p_{22} & p_{23} \\ p_{11} & p_{12} & p_{13} \end{bmatrix}, \qquad P_2 = \begin{bmatrix} p_{33} & p_{32} & p_{31} \\ p_{23} & p_{22} & p_{21} \\ p_{13} & p_{12} & p_{11} \end{bmatrix}.$$

The matrix P_2 can also be achieved if the columns of P are flipped first, and the rows of the resulting matrix are flipped afterwards [38]. The (i, j) element in matrix G is calculated by placing the centre of P_2 on each (i, j) element of F.
The convolution with the 2D PSF is an LSI, which allows the blurred image $\mathcal{G}$ to be obtained. In particular, each pixel in $\mathcal{G}$ is influenced by the neighbouring of its corresponding pixel in $\mathcal{F}$. For example, b_{22} is measured as the weighted

combination of the PSF and the neighbouring pixels near to a_{22}, a_{12}, a_{23}, a_{32}, and a_{21}. However, there are some pixels with neighbouring outside the border of $\mathcal{F}$, and therefore an assumption of the boundary condition should be considered. The following representation of $\mathcal{F}$ considers the scene outside the border:

w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
w_{21}	a_{11}	a_{12}	a_{13}	w_{25}
w_{31}	a_{21}	a_{22}	a_{23}	w_{35}
w_{41}	a_{31}	a_{32}	a_{33}	w_{45}
w_{51}	w_{52}	w_{53}	w_{54}	w_{55}

(2.9)

where w_{ij} denotes the elements at the border of F.

The structure of the blurring matrix H is dependent on the assumed following boundary conditions.

Zero boundary condition [38]. If this condition is assumed, then the elements w_{ij} are equal to zero. The element b_{22} of G is given by overlapping F and P_2, such that the elements a_{22} and p_{22} match:

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

while the element b_{21}, located at the border of G, is given by overlapping F and P_2, such that the elements a_{21} and p_{22} are matched:

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

Similarly, the calculations are repeated for all other elements of G. It follows, by using Eq. 2.2 when $\mathbf{n} = 0$, that

$$\begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{13} \\ b_{23} \\ b_{33} \end{bmatrix} = \left[\begin{array}{ccc|ccc|ccc} p_{22} & p_{12} & & p_{21} & p_{11} & & & & \\ p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} & & & \\ & p_{32} & p_{22} & & p_{31} & p_{21} & & & \\ \hline p_{23} & p_{13} & & p_{22} & p_{12} & & p_{21} & p_{11} & \\ p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} \\ & p_{33} & p_{23} & & p_{32} & p_{22} & & p_{31} & p_{21} \\ \hline & & & p_{23} & p_{13} & & p_{22} & p_{12} & \\ & & & p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} \\ & & & & p_{33} & p_{23} & & p_{32} & p_{22} \end{array}\right] \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}.$$

Periodic condition . It is assumed here that the image $\mathcal{F}$ is repeated endlessly in all directions, so that elements w_{ij} in Eq. (2.9) show as the following form:

$$\left[\begin{array}{c|cccc} a_{33} & a_{31} & a_{32} & a_{33} & a_{31} \\ \hline a_{13} & a_{11} & a_{12} & a_{13} & a_{11} \\ a_{23} & a_{21} & a_{22} & a_{23} & a_{21} \\ a_{33} & a_{31} & a_{32} & a_{33} & a_{31} \\ \hline a_{13} & a_{11} & a_{21} & a_{31} & a_{11} \end{array}\right] \tag{2.10}$$

which, by using Eq. 2.2 when $\mathbf{n} = 0$, yields

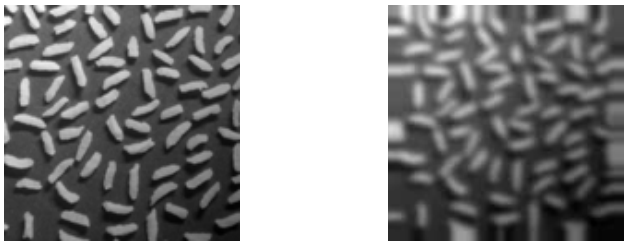
$$\begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{13} \\ b_{23} \\ b_{33} \end{bmatrix} = \left[\begin{array}{ccc|ccc|ccc} p_{22} & p_{12} & p_{32} & p_{21} & p_{11} & p_{31} & p_{23} & p_{13} & p_{33} \\ p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} & p_{33} & p_{23} & p_{13} \\ p_{12} & p_{32} & p_{22} & p_{11} & p_{31} & p_{21} & p_{13} & p_{33} & p_{23} \\ \hline p_{23} & p_{13} & p_{33} & p_{22} & p_{12} & p_{32} & p_{21} & p_{11} & p_{31} \\ p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} & p_{31} & p_{21} & p_{11} \\ p_{13} & p_{33} & p_{23} & p_{12} & p_{32} & p_{22} & p_{11} & p_{31} & p_{21} \\ \hline p_{21} & p_{11} & p_{31} & p_{23} & p_{13} & p_{33} & p_{22} & p_{12} & p_{32} \\ p_{31} & p_{21} & p_{11} & p_{33} & p_{23} & p_{13} & p_{32} & p_{22} & p_{12} \\ p_{11} & p_{31} & p_{21} & p_{13} & p_{33} & p_{23} & p_{12} & p_{32} & p_{22} \end{array}\right] \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}.$$

Reflexive condition [38]. The blurring matrix H, in this case, is structured as the sum of the structured matrices BTTB, BHTB, BTHB, and BHHB [38]. The entries w_{ij} in F are represented as:

$$\begin{array}{c|cccc} a_{11} & a_{11} & a_{12} & a_{13} & a_{13} \\ \hline a_{11} & a_{11} & a_{12} & a_{13} & a_{13} \\ a_{21} & a_{21} & a_{22} & a_{23} & a_{23} \\ a_{31} & a_{31} & a_{32} & a_{33} & a_{33} \\ \hline a_{31} & a_{31} & a_{32} & a_{33} & a_{33} \end{array} \tag{2.11}$$

Figure 2.6 (c) shows the effect of convolving a clean image in (a) with a Gaussian PSF of size 15×15 using the reflexive boundary condition.

The Gaussian PSF is a typical example of a separable blur, namely, the PSF can be decomposed into horizontal and vertical components [78]. The convolution computation with a separable PSF is discussed in the next section.



(a) **(b)**

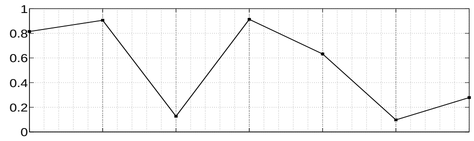
Figure 2.6: *An example of the convolution in 2D problem with reflexive boundary condition.*

2.5.2 Convolution by separability

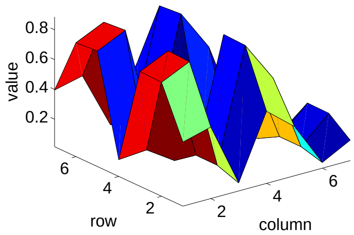
The convolution can be achieved quickly if a given PSF array is found to be separable. If the PSF can be decomposed into two column vectors, one representing its horizontal components and the other representing its vertical components, then the PSF is said to be separable. Mathematically, given a matrix P of a separable PSF, the following holds:

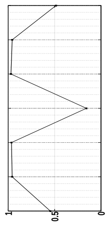
$$P = \mathbf{P}_c \mathbf{P}_r^T = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \begin{bmatrix} r_1 & r_2 & \dots & r_n \end{bmatrix}, \tag{2.12}$$

where c_i and r_j are the components of column vector $\mathbf{P}_c$ and row vector $\mathbf{P}_r$, for $i = 1, \dots, m$ and $j = 1, \dots, n$, respectively [38, 78]. The same result is obtained if the form $P = \mathbf{P}_r \mathbf{P}_c^T$ is used. The rank of any separable PSF is always equal to 1 [38], otherwise it is a non-separable PSF. The elements in the blurring matrix H that contains the information of separable PSF is given as $p_{ij} = c_i r_j$.



0.81	0.90	0.13	0.91	0.63	0.10	0.28

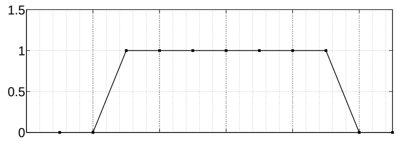




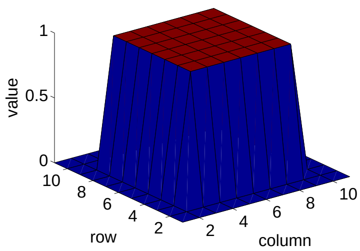
0.55
0.96
0.96
0.16
0.97
0.96
0.48

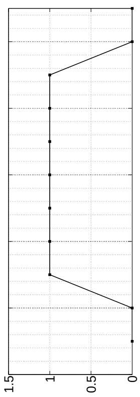
0.45	0.49	0.07	0.50	0.34	0.05	0.15
0.78	0.88	0.12	0.87	0.60	0.09	0.27
0.79	0.87	0.12	0.88	0.61	0.09	0.27
0.13	0.14	0.02	0.14	0.10	0.01	0.04
0.79	0.88	0.12	0.89	0.61	0.09	0.27
0.78	0.87	0.12	0.87	0.60	0.09	0.27
0.39	0.44	0.06	0.44	0.31	0.05	0.13

Figure 2.8: *An example of a separable 2D PSF.*



0	0	1	1	1	1	1	1	1	0	0





0
0
1
1
1
1
1
1
1
1
1
0
0

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Figure 2.7: *An illustration of a simple separable square PSF.*

Figure 2.7 provides an illustration of separability using a simple square PSF of size 11×11. Figure 2.8 shows an example of a random separable PSF in 2D, such that the PSF is decomposed into two 1D arbitrary array as shown in the diagram. This diagram contains the numerical entries of the PSF; different representation are shown in the left.

Given a blurred image, a reverse procedure of the convolution is required to restore the image; this is referred to as a deconvolution. Having discussed how the convolution procedure is performed, the next section introduces some of the main approaches proposed to the image deconvolution problem.

2.6 Image restoration methods

An overview of image restoration fundamentals was given in the previous sections. This section reviews some of the most frequently used image restoration methods. Several methods have been proposed to restore distorted images; however, they mostly require advance information about the problem. The image deconvolution techniques can be classified into two main approaches, where the determination of the PSF and the true image are performed either separately or simultaneously [51].

The first approach identifies the the blurring function separately from the exact image [51]. Any *a priori* blur identification technique is classified under this approach, i.e. it is mainly based on the PSF determination, regardless of the true image restoration. This approach is commonly used in many applications such as astronomy [22], photography and video [29, 81], and medical imaging [74]. It

is built on the assumption that the true image holds some features such as point sources or edges; moreover, the PSF is assumed to be symmetric with a possibly known parametric form of the blur [51]. The blur can be estimated using different techniques, such as spectral blur estimation [33], while the true image can be obtained using the classical methods of image restoration [33], including inverse filtering [7], Weiner filter [35], least squares and constrained least squares filters, and iterative methods [7, 53], after identifying the PSF. The complexity of the linear approach is considered to be very low compared to the non-linear approach[51]. Furthermore, this approach can solve computationally simple algorithms [51].

The second approach incorporates the identification procedure with a restoration algorithm, where the determination of the PSF and the true image are simultaneous procedures [51]. One such method is called zero sheet separation; it refers to the process of factoring the z-transform of the 2D blurred image [70, 76]. Methods using this approach are claimed to be the fastest deblurring algorithm; it is based on many assumptions such as the irreducibility of both the true image and the PSF, with a finite support of the PSF, and the absence of noise [51]. Furthermore, methods using this approach are considered to be very sensitive to additive noise. Moreover, the second approach is used to solve more complex algorithms [51]. Other methods based on the second approach have been studied and reviewed in [51], including the non-parametric deterministic constraints algorithm [50], the high order statistic HOS algorithms [63], and the autoregressive moving average ARMA parameter estimation algorithm including maximum likelihood blur estimation [54], regularised filtering and cross validation methods [38, 50, 71], and the Richardson-Lucy method [89]. Spectral methods are also used for the blur

identification [16]. On the other hand, several methods based on algebraic geometry techniques have been implemented to solve the BID problem [24, 60, 69]. Some of the most frequently used image restoration methods are discussed in the following sections. Non-blind image deconvolution methods based on *a priori* observations of the PSF will be introduced first, including the inverse filter in Section 2.6.1, the Wiener filter in Section 2.6.2, the constrained least squares method in Section 2.6.3, and the iterative methods in Section 2.6.4. Finally, the blind maximum likelihood restoration method will be presented in Section 2.6.5.

2.6.1 Inverse filter

As mentioned above, the inverse filter is considered to be one of the simplest approaches to solve the deconvolution problem [8, 33]. It refers to the naive solution of Eq. 2.2, which aims to find an approximate $\hat{\mathbf{f}}$ of the exact image $\mathbf{f}$ by inverting the blurring matrix H. In noiseless conditions, this can be modelled by

$$\mathbf{f} = H^{-1}\mathbf{g}_{exact}. \tag{2.13}$$

where $\mathbf{g}_{exact}$ denotes the column vector representation of the blurred image $\mathcal{G}$ in the noiseless condition. In the presence of noise, in particular ($\mathbf{g} = \mathbf{g}_{exact} + \mathbf{n}$), the inverse solution of Eq. 2.2, is given by

$$\begin{aligned} \hat{\mathbf{f}} &= H^{-1}\mathbf{g}, \\ &= H^{-1}\mathbf{g}_{exact} + H^{-1}\mathbf{n}, \\ &= \mathbf{f} + H^{-1}\mathbf{n}. \end{aligned} \tag{2.14}$$

The inverse filter uses spectral components of the blurred image to find the solution. However, the inverse filter is not appropriate when some values of H are equal to zero, and, introducing noise implies that the solution $\hat{\mathbf{f}}$ will be dominated by noise.

It is important to define the singular values decomposition (SVD), to deal with the inverted noise $H^{-1}n$. As noted in the literature, the SVD is a tool used frequently for analysing the numerical stability of a system.

Definition 2.2. Suppose $A \in \mathbb{R}^{m\times n}$ is a matrix of size $m \times n$ in the field of real numbers $\mathbb{R}$. Then there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$and $V \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} A &= U\Sigma V^T, \quad \Sigma \in \mathbb{R}^{m\times n}, \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}, \\ \Sigma_1 &= \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k), \qquad k = \min(m, n), \end{aligned}$$

where σ_i for $i = 1, \ldots, k$ are the singular values of A such that:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq \sigma_{r+1} = \ldots = \sigma_k = 0.$$

The number of non-zero singular values σ_i of the matrix $A \in \mathbb{R}^{m\times n}$ is equal to its rank [11, 31]. However, in the presence of round-off error and noisy data the rank determination becomes a non-trivial task.

It follows that the singular values of the blurring matrix H can be represented

as

$$\begin{aligned} H &= U\Sigma V^T \\ &= \begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} \\ &= u_1\sigma_1 v_1^T + \ldots + u_k\sigma_k v_k^T \\ &= \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \end{aligned} \tag{2.15}$$

where $\mathbf{u}_i, \mathbf{v}_i$, and σ_i are the columns vector of the orthogonal matrices U and V, and the nonzero values in the diagonal matrix Σ respectively [38]. Likewise, the inverted matrix of H is decomposed by SVD:

$$H^{-1} = \sum_{i=1}^{k} \frac{1}{\sigma_i} \mathbf{u}_i^T \mathbf{v}_i. \tag{2.16}$$

Exploiting the relation between Eqs. 2.13 and 2.16, an approximation of $\hat{\mathbf{f}}$ thus can be modelled as

$$\hat{\mathbf{f}} = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{g}}{\sigma_i} \mathbf{v}_i. \tag{2.17}$$

Likewise, the inverted noise $H^{-1}n$ in Eq. 2.14 is given by

$$H^{-1}n = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T n}{\sigma_i} \mathbf{v}_i.$$

As was pointed out earlier, the matrix H is badly ill-conditioned because a small bounded error in the input can result in a large unbounded error in the output [8]. Therefore, solving the problem in presence of noise does not give an adequate estimation to the exact image, because of the inverted noise $H^{-1}n$ [8, 33, 38]. The spectral components $|\mathbf{u}_i^T \mathbf{g}|$, which correspond to the larger singular values, contain the exact data and represent the lower frequency information, but the

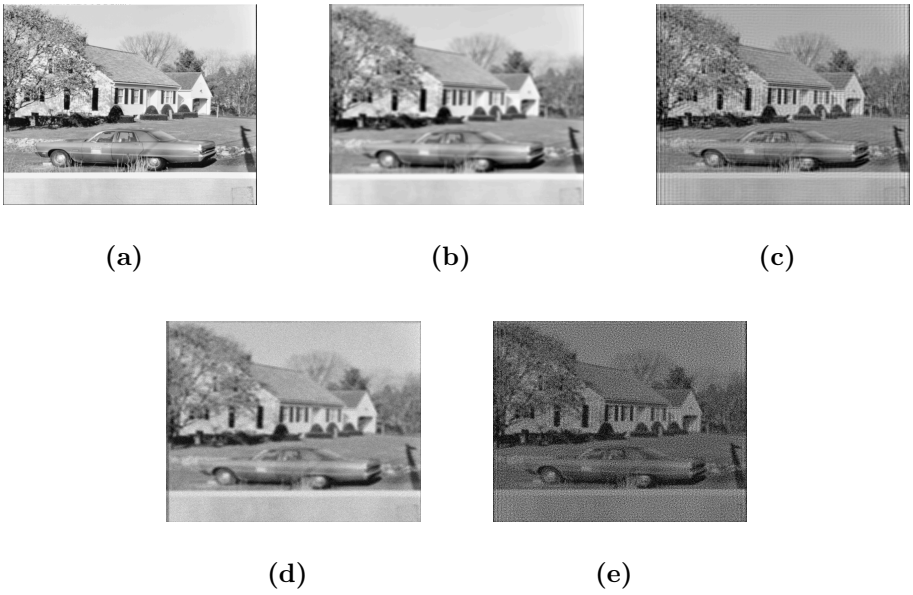


Figure 2.9: *Image restoration using the inverse filter.* (a) an exact image; (b) a blurred image; (c) a restored image of (b); (d) a blurred image with noise; and (e) a restored image of (d)

smaller components are corrupted by the additive noise.

Figure 2.9 compares the inverse filter results, when an exact image that is shown in (a) is distorted by a blur only as shown in (b), and when it is distorted by the blur plus an additive noise as shown in (d). The noiseless condition yields a sharper image, as shown in (c), while the inverse filter failed to show a similar result when the noise is introduced, as can be seen in (e). Since the inverse filter method behaves badly with additive noise, a better solution for this problem may involve regularisation methods or the least squares method.

Regularisation methods define a certain parameter that can reduce the effect of noise in the restored image. In particular, the small components in $\hat{\mathbf{f}}$ Eq. 2.17 contain small singular values that require to be filtered, as they are dominated by noise [71]. These regularisation methods can be represented using the solution

in Eq. 2.17, by defining a filter factor ϕ_i such that

$$\hat{\mathbf{f}}_{filt} = \sum_{i=1}^{k} \phi_i \frac{\mathbf{u}_i^T \mathbf{g}}{\sigma_i} \mathbf{v}_i.$$

Two methods of regularisation are considered in [38] in order to determine the filter factor ϕ_i. The first is truncated singular value decomposition (TSVD), and the second is Tikhonov regularisation. In the TSVD method, the SVD components corresponding to the smallest singular values are set to be zero, so that

$$\phi_i = \begin{cases} 1 & i = 1, \ldots, d, \\ 0 & i = d+1, \ldots, k. \end{cases} \qquad 1 \leq d \leq k,$$

where d denotes the truncation parameter [38]. The Tikhonov method computes the approximation $\hat{\mathbf{f}}$ using the following minimisation problem:

$$\min_{\mathbf{f}} \left\{ \|\mathbf{g} - H\mathbf{f}\|_2^2 + \mu^2 \|\mathbf{f}\|_2^2 \right\},$$

where μ denotes the regularisation parameter. This parameter can be determined using the filter factor ϕ_i

$$\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \mu^2}, \qquad i = 1, \ldots, k.$$

To be more precise, the parameter μ defines the level at which the σ_i come down due to noise in the image. In order to obtain an acceptable filter factor ϕ_i the discrete Picard condition must be satisfied. This is satisfied if, for all σ_i larger than μ, the corresponding coefficients decay faster than the σ_i [36]. The TSVD filters out the smaller singular values with a sharp cut-off, while Tikhonov filters out the smaller singular values with a gradual roll-off, and TSVD and Tikhonov methods yield similar results. The regularisation parameter can be defined manually; also it can be computed automatically using three methods: the generalised cross validation [32, 71], the L-curve criterion [37], and the discrepancy

principle [38]. Moreover, the least squares methods can deal with the noise better than inverse filter.

The next section discusses one of the least squares filters, the Wiener filter.

2.6.2 Wiener filter

The inverse filter method is very sensitive to additive noise, and therefore a method called Wiener filter has been proposed to address this deficiency in the image restoration. The Wiener filter is one of the linear approaches based on the least squares method; it requires an *a priori* statistical knowledge of the exact image $\mathbf{f}$, noise $\mathbf{n}$, and a model for the minimisation problem which may given by [8]:

$$\arg_{\hat{\mathbf{f}}} \min \|\mathbf{f} - \hat{\mathbf{f}}\|_2^2. \tag{2.18}$$

More precisely, the Wiener filter uses the LSI system where the determination of the blurring matrix H of the PSF requires to minimise the MSE, between $\mathbf{f}$ and the estimated image $\hat{\mathbf{f}}$ [54]. The Wiener filter performs two procedures simultaneously on the blurred image, first inverting the blurring filter H and then reducing the noise (i.e. noise smoothing).

The solution of Eq. 2.18 is expressed in spectral domain by

$$H_w = \frac{H^*(u,v) S_{\hat{f}}(u,v)}{|H(u,v)|^2 S_{\hat{f}}(u,v) + S_n(u,v)}, \tag{2.19}$$

where H^*, $S_{\hat{f}}$ and S_n are the complex conjugate of H, the power spectra of $\hat{\mathbf{f}}$ and $\mathbf{n}$ respectively, and u and v are the components in spectral domain. The power spectra represent the distribution of the average signal power and noise power among its frequency units in the image [54]. The Wiener filter performs the deconvolution using an inverse filter and a compression procedure to remove

the additive noise for each frequency, implying $H(u, v) = 0$, mainly by using the low-pass filter [54]. When no additive noise is introduced to the problem only the deconvolution process is performed. The implementation of the Wiener filter requires an estimation of the power spectra of both the exact image and the noise. In particular, if white noise is introduced then S_n is a measure as the variance of the noise, such that

$$S_n(u, v) = \sigma_n^2, \tag{2.20}$$

for each spatial component (u, v). If the values of σ_n^2 are very small then the restored image will be similar to the one obtained by inverse filter, and it will be oversmoothed if the values of σ_n^2 are large [54].

The estimator of the power spectra of the exact image $S_f(u, v)$ can be done using several methods. One direct method is called the periodogram estimator, where $S_f(u, v)$ is computed from the blurred image as

$$\begin{aligned} S_f(u, v) &\approx S_g(u, v) - S_n(u, v), \\ S_g(u, v) &= \frac{1}{mn} g(u, v) g^*(u, v), \end{aligned}$$

where $g^*(u, v)$ is the complex conjugate of the blurred image $\mathcal{G} \in \mathbb{R}^{m \times n}$, [54].

An estimator of $S_f(u, v)$ can be obtained from a set of images that are similar to the exact image. Other methods include some statistical models, such as the autoregressive model [54].

(a) Clear image **(b)** Distorted image **(c)** Restored image

Figure 2.10: *Image restoration using Wiener filter.* (a) an exact image; (b) a blurred noisy image; and (c) the restored image.

It can be easily seen from Figure 2.10 that the image details have been improved but the noise has amplified. The Wiener filter was implemented using exactly known PSF, on a blurred image contaminated with a random noise of 10^{-3}.

In the next section another type of least squares filter will be discussed.

2.6.3 Constrained least squares filtering

This section presents a method of image restoration that is similar to the Wiener filter in how it deals with the noise amplification problem which arises from using the inverse filter. Another limitation arises as the Wiener filter requires an estimation of the power spectra of the exact image, so some prior information is needed. The constrained least squares filtering method is proposed to deal with these limitations [3, 31, 43]. Basically, it assumes that the blurred image of the restored and the exact images are almost equal [54]. More precisely, if the following model is assumed:

$$H\hat{\mathbf{f}} \approx \mathbf{g},$$

then

$$\|\mathbf{g} - H\hat{\mathbf{f}}\|^2 = \sigma_n^2, \tag{2.21}$$

is satisfied [54]. Since there are several possible solutions of Eq. 2.21, a second criterion is assumed based on the smoothness of the solutions. In particular, the second criterion is subject to the minimisation problem in Eq. 2.21. The solution to this minimisation problem leads to an estimate of the exact solution in the Fourier domain provided by

$$\hat{\mathbf{f}} = \Big[\frac{H^*(u,v)}{|H(u,v)|^2 + \alpha|P(u,v)|^2}\Big] g(u,v), \tag{2.22}$$

where $P(u,v)$ is the Fourier transform of the Laplacian function

$$P = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \tag{2.23}$$

and α is a regularisation parameter such that Eq. 2.21 is accomplished. The Laplacian operator is a smoothing filter, that is implemented here to identify the fine detail in the blurred image, in particular, it is used to remove noise [33]. The ringing effect can be present in the restored image; when the value of α is large. If the value of α is small, the effect of amplified noise in the restored image is present; this is referred to as an undersmoothed image [3, 38, 54]. The regularisation parameter can be estimated using several methods, as described in Section 2.6.1.

The following section introduces the iterative methods.

2.6.4 Iterative methods

Many efficient methods of image restoration can be solved using spectral filtering, but only for simple LSI systems as discussed in the last two sections. In many methods convolution in the spatial domain is chosen over convolution in the spectral domain when a given image is too large [57], and when some required prior knowledge cannot be expressed appropriately in the spectral domain, for instance the positivity of image intensities [57]. In particular, the positivity in some image restoration methods based on the spectral domain, such as the Wiener filter and the constrained least squares filter [54], is not restricted. Many iterative methods based on the spatial domain are investigated in [7, 53, 57].

Basically, iterative restoration methods preform an iteration procedure on the solution obtained by using the inverse filter. The method is modelled by the following formula:

$$\hat{\mathbf{f}}_{k+1} = \hat{\mathbf{f}}_k + \beta(\mathbf{g} - H\hat{\mathbf{f}}_k), \qquad k = 0, \cdots, c, \tag{2.24}$$

where β is the converge parameter and k is the number of iterations such that the solution is converged after c iterations. The parameter β satisfies

$$|1 - \beta H| < 1, \qquad \text{for each spatial component,} \tag{2.25}$$

with the assumption that $|H| \leq 1$, the Eq. 2.25 satisfies

$$0 < \beta < 2 \qquad \text{and} \qquad H > 0. \tag{2.26}$$

It can be easily seen that if $k = 0$ in Eq. 2.24, the solution becomes equal to the blurred image that corresponds to $\mathbf{g}$. This form is called the Landweber iteration and is used in most iterative methods, including Richardson-Lucy [89]. The solution in each iteration $\hat{\mathbf{f}}_k$ is compared with the $\mathbf{g}$ of the blurred image, and the

output is used then to update the output of the next iteration. In particular, too many iterations of Eq. 2.24, defined by the parameter β, will lead to an inverse filtered image where the additive noise is amplified as the image deblurs [54]. A relevant solution is considered when an accurate value of the parameter β is defined [54].

More constraints can be added to Eq. 2.24, to allow the use of *a priori* knowledge, such as the positivity of the image intensities, as mentioned earlier. So by eliminating all the negative intensities and replacing with zeros, Eq. 2.24 can be written as

$$\hat{\mathbf{f}}_{k+1} = \boldsymbol{P}\left[\hat{\mathbf{f}}_k + \beta(\mathbf{g} - H\hat{\mathbf{f}}_k)\right], \qquad k = 0, \cdots, c, \tag{2.27}$$

where $\boldsymbol{P}$ is a projection operation [54]. Some works have explored different definitions of the parameter β and the projection $\boldsymbol{P}$ [3]. The iterative restoration method is extended to include models with spatially variant PSF [48, 54].

There are some disadvantages to using this method, including the assumption of $H > 0$ in Eq. 2.26 for each spatial component, which is not satisfied in practice [54], and the lack of information in the spectral domain [54]. To overcome these difficulties a combination of the iterative methods in Eq. 2.24 and the constrained least squares filter is considered [54]. Another drawback to this method is that the iterative methods have a slow convergence rate; a better solution may involve a conjugate gradient algorithm [57].

Richardson-Lucy is a well-known simple iterative deconvolution method [72]. A clearer image is obtained using similar iteration procedures; however, ringing effects commonly appear on the boundaries of the objects of the restored image. Several works propose an enhancement of the Richardson-Lucy method, to reduce

the ringing effects, basically by assuming the dissimilarity between the edge and the smooth area of the image [89].

(a) Clear image **(b)** Blurred image

(c) Iterative filter **(d)** Richardson-Lucy

Figure 2.11: *The image restoration using the iterative filters and Richardson-Lucy deconvolution method, in presence of noise.*

Figure 2.11 shows an example of image restoration of a blurred image with additive noise shown in (b), using the described iterative method in (c) such that the iteration is 200, and the Richardson-Lucy method in (d). Obviously, the restored image in (c) becomes darker than the blurred image; however, it contains many more detail and the result is similar to that of the Richardson-Lucy method. It also can be noticed that the inverse filter may have an advantage over the iterative filter in noiseless conditions, while the iterative filter is more efficient when noise is introduced to the blurred image.

2.6.5 Maximum likelihood blur estimation methods

This method is an enhanced technique of the iterative constrained algorithm. It is based on a statistical model and aims to optimise an approximate $\hat{\mathbf{f}}$ of the exact $\mathbf{f}$ iteratively. The estimation process of $\hat{\mathbf{f}}$ and the elements of the PSF are performed simultaneously.

In most of the image restoration methods, a model of the exact image and knowledge of the PSF and the additive noise are required. Recall from Section 2.6.2 that the power spectra of an exact image can be estimated using a statistical model. This involves the 2D autoregressive (AR) model, given by

$$f(x,y) = a_{01}f(x,y-1) + a_{11}f(x-1,y-1) + a_{01}f(x-1,y) + \nu(x,y), \quad (2.28)$$

where a_{ij} are the AR coefficients, and $\nu(x,y)$ is an unpredictable spatial component [54]. The parameters a_{ij} and $\nu(x,y)$, with variance σ_ν equal to white noise, are assumed to be unknown. In this case a function called log likelihood is established to estimate a set of parameters θ from the given distorted image, such that $\theta = \{a_{ij}, \sigma_\nu, \sigma_n, H\}$ [55, 56]. The problem can be solved in the spatial domain; however, some works consider that the spectral domain has an advantage [56]. The log likelihood equation for Gaussian distribution is given by [54]

$$L_\theta = \sum_{u,v} \Big(\log P(u,v) + \frac{g(u,v)g^*(u,v)}{P(u,v)} \Big),$$

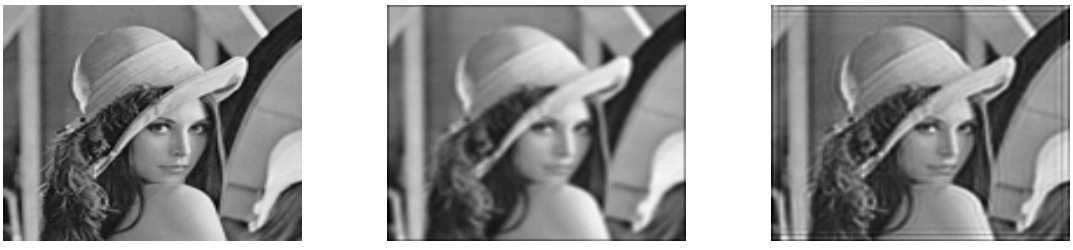
where $P(u,v)$ is the probability density function (PDF) of the blurred image. The PDF is obtained by combining the image restoration model in Eq. 2.2 and

the AR model in Eq. 2.28, leading to

$$P(u,v)|_{\mathbf{g},\theta} = \sigma_\nu^2 \frac{|H(u,v)|^2}{|1 - A(u,v)|^2} + \sigma_n^2. \tag{2.29}$$

where $A(u,v)$ represents the coefficients a_{ij} in a 2D discrete Fourier transform. Image restoration using the maximum likelihood blur estimation method is achieved by estimating the parameters of θ that maximise the log likelihood function [49]. The main limitation of this method is that more constraints should be applied in order to obtain an acceptable estimation of the PSF, such that its sum is equal to 1 [55]. Another limitation is found in the log likelihood function itself: it is excessively a non-linear optimisation method, and no unique solution is available [55].

This development in the blur estimation using maximum likelihood algorithm with unknown PSF continuously improves the image restoration [42, 48, 49]. It involves a better initialisation of the required parameters, such as in the expectation-maximization algorithm [55, 56]. Figure 2.12 shows the image restoration using maximum likelihood deconvolution, which gives a similar result to the Richardson-Lucy deconvolution method.



(a) Original image **(b)** Blurred and noisy image **(c)** Restored image

Figure 2.12: *Blind image deconvolution using maximum likelihood methods.*

2.7 Summary

This section has reviewed the basic concepts of image restoration including image formation, the blurring function known as the PSF, and image convolution. Having discussed some of the main existing approaches of image restoration, the following chapter will present important tools used in the new proposed method of image restoration.

Chapter 3

The GCD approach and image deblurring

3.1 Introduction

It was seen that, the BID problem is an example of ill-posed linear problem [57, 69]. This problem was modelled in Eq. 2.2, where H is an ill-conditioned matrix that contains the PSF information. The determination of the original image requires the calculation of the vector $\mathbf{f}$. The main aim is to remove the blur from the input image $\mathbf{g}$, where it is considered to be more significant than the noise $\mathbf{n}$ [57]. The solution of this problem is connected to the computation of the GCD of two bivariate polynomials whose coefficients are exactly known. Several algorithms have been established to retrieve the original image from a single image or from many distorted images, using the GCD approach [24, 60, 69]. Thus, the BID problem is an application of GCD computations, where the PSF array, the true and degraded images are defined as bivariate polynomials. The PSF ar-

ray is considered as the GCD of two dissimilar true images, such that each image contains the true PSF. Before proceeding further with the BID problem, it is necessary to define the GCD of two exact univariate polynomials and an AGCD of two inexact univariate polynomials. This chapter provides an overview of the GCD approach and its approximations of AGCDs of two univariate polynomials, which will be introduced in Sections 3.2 and 3.3 respectively. Some well-known approaches for AGCD computation are addressed in Section 3.4.

3.2 Definition of the GCD of a pair of exact polynomials

The calculation of the GCD of two polynomials is a basic approach in the field of algebraic computation.

Definition 3.1. Let $f_1(x)$ and $f_2(x)$ be two univariate polynomials whose coefficients are known exactly:

$$f_1(x) = \hat{a}_m x^m + \hat{a}_{m-1} x^{m-1} + \hat{a}_{m-2} x^{m-2} + \cdots + \hat{a}_0, \tag{3.1}$$

$$f_2(x) = \hat{b}_n x^n + \hat{b}_{n-1} x^{n-1} + \hat{b}_{n-2} x^{n-2} + \cdots + \hat{b}_0, \tag{3.2}$$

where $\hat{a}_0, \hat{a}_1, \hat{a}_2, ..., \hat{a}_m$ and $\hat{b}_0, \hat{b}_1, \hat{b}_2, ..., \hat{b}_n$ are the coefficients of $f_1(x)$ and $f_2(x)$ respectively and $a_m, b_n \neq 0$, and m and n represent the degree of polynomials $f_1(x)$ and $f_2(x)$ respectively.

The GCD of two univariate polynomials $f_1(x)$ and $f_2(x)$ is defined as a polynomial

$\hat{d}(x)$ of highest degree that can divide $f_1(x)$ and $f_2(x)$:

$$\begin{aligned} \hat{d}(x) &= \text{GCD}\,(f_1(x), f_2(x)), \\ \hat{d}(x) &= \frac{f_1(x)}{\hat{u}(x)} = \frac{f_2(x)}{\hat{v}(x)}, \end{aligned}$$

where $\hat{u}(x)$ and $\hat{v}(x)$ are referred to as the quotient polynomials, and are co-prime.

In many applications, the computational process of the GCD can produce a very large number of errors leading two exact (i.e. error-free) polynomials which have a non-constant GCD to become co-prime. If an arbitrary small perturbation appears in the coefficients of $f_1(x)$ and $f_2(x)$, it will make them co-prime polynomials. As a result of this deficiency of GCD computations, an AGCD is introduced. In the next section, the most well-known definitions of AGCD are presented and one of them is assumed.

3.3 Defnition of an AGCD of a pair of inexact polynomials

A tremendous amount of literature has been published on the AGCD of a pair of univariate polynomials whose coefficients are inexact. These studies have given several definitions of the AGCD as well as many methods for computing one. An AGCD arises in many applications including, for example, polynomial root finding [64], computer vision [28], image deblurring [1, 51, 60, 69, 83], control theory [4], system identification [80], and hybrid rational approximation [44]. The GCD of a pair of exact polynomials $f_1(x)$ and $f_2(x)$ yields a unique result $\hat{d}(x)$, while the

AGCD of two inexact polynomials $g_1(x)$ and $g_2(x)$ dose not have a unique result. In particular, different definitions of an AGCD yields different formation of the problem. Some studies [9, 20, 26, 75] have used the definition in [66] for ε-GCD of polynomials, where the degree of an ε-GCD is uniquely defined, but not its coefficients. The definition of an ε-GCD is formalised as follows.

Definition 3.2. [66], Let g_1 and g_2 be two univariate polynomials of degrees m and n respectively within error bound $\varepsilon > 0$ and a given norm in the space of polynomials. Then the ε-GCD is defined when:

- There exist perturbed polynomials $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$ of g_1 and g_2, respectively:

$$\begin{aligned} \tilde{g}_1(x) &= g_1(x) + \delta g_1(x), \qquad \|\delta g_1\| \leq \varepsilon, \\ \tilde{g}_2(x) &= g_2(x) + \delta g_2(x), \qquad \|\delta g_2\| \leq \varepsilon, \end{aligned} \tag{3.3}$$

  that satisfy:

  1. The polynomial ε-GCD is equal to $\mathrm{GCD}(\tilde{g}_1(x), \tilde{g}_2(x))$.
  2. The degree of ε-GCD is equal to the maximum integer r such that:

$$\deg(\mathrm{GCD}\,(\tilde{g}_1(x), \tilde{g}_2(x))) = r.$$

However, in [47] the definition of an AGCD is presented as 'the nearest GCD' problem, and formalised as follows.

Definition 3.3. Let g_1 and g_2 be two polynomials of degrees m and n respectively, with a given norm in the space of polynomials. Then the nearest-AGCD is defined as the nearest $\tilde{g}_1(x)$, and $\tilde{g}_2(x)$ to g_1 and g_2 respectively, such that:

$$\|g_1(x) - \tilde{g}_1(x)\|^2 + \|g_2(x) - \tilde{g}_2(x)\|^2, \tag{3.4}$$

is minimised. Moreover, $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$ have a non-trivial GCD.

One of the limitations with ε-GCD is that it fails to consider the uniqueness of the exact GCD [39], since there are many polynomials $\tilde{g}_1(x)$, and $\tilde{g}_2(x)$ that are close to g_1 and g_2 and are bounded by error $\varepsilon > 0$. Moreover, ε-GCD suffers from an instability problem [65], where small perturbations in the coefficients of the polynomials $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$ produce different degrees. To overcome this problem, Pan [65] has introduced a new definition δ-GCD based on initial computation of polynomial roots. δ-GCD has a limitation because it requires an accurate computation of its roots. On the other hand, Zeng [93] summarises the characteristics of an AGCD around three properties: 'nearness' as defined in Definition 3.3, and 'maximum degree' and 'minimum distance' as defined in 3.2. In general, a better definition of an AGCD is necessary to accommodate all these properties, and therefore Definition 3.4 will be assumed.
One aim of this research is to obtain an appropriate definition of the following problem: Given a pair of polynomials g_1 and g_2, find the degree of an AGCD, perturbations $\delta f(x)$ and $\delta g(x)$, and an AGCD polynomial. Therefore, the following definition of an AGCD is assumed in this research.

Definition 3.4. [39]. Let g_1 and g_2 be a pair of inexact polynomials of degrees m and n respectively. The polynomial $\tilde{d}(x)$ with degree d is said to be an AGCD of the perturbed polynomials $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$, if the following properties are satisfied.

1. The polynomials $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$ are of degrees m and n respectively.

2. The distance in Eq. 3.4 is minimised.

3. The degree of $\tilde{d}(x)$ is equal to the rank loss of the Sylvester resultant matrix of $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$.

This definition assumes that the degree of an AGCD is computed using the methods provided in [84].

Several studies use earlier definitions of an AGCD of univariate polynomials, for example the quasiGCD method [5] uses Euclid's algorithm with a look-ahead algorithm, the QRGCD method [21] is based on the QR decomposition of Sylvester resultant matrix, and the uvGCD algorithm [93] uses a black box-type algorithm and the QRGCD approach to increase accuracy in the performance.

These approaches are discussed in the following section.

3.4 Previous works on AGCD computations

The computation of an AGCD of a pair of inexact polynomials can be performed using several approaches, after which a definition of an AGCD is selected. In particular, this research will follow the AGCD in Definition 3.4 which is based on the ε-GCD definition. This section introduces three of the main computational approaches that have been used to compute the AGCD of two inexact univariate polynomials: Euclid's algorithm, the resultant and subresultant matrices approach, and the optimisation techniques approach.

Euclid's algorithm has been used to find the GCD of two exact polynomials. In addition, several researchers [5, 14, 19, 41, 64, 77] have extended Euclid's algorithm to the computation of AGCD of two inexact polynomials. The GCD computation of two univariate polynomials whose coefficients are exact is based on the classical version of Euclid's algorithm [12]. Euclid's algorithm is very ef-

ficient with exact coefficients, but problems arise with inexact coefficients. This algorithm fails with AGCD computation, because it fails to terminate. In general, most of the AGCD computation studies based on Euclid's algorithm have considered this limitation and suggested some modifications to the main algorithm. For instance, Hribernig and Setter [41] have modified the termination criteria, while Noda [64] extended the algorithm to an approximate square-free decomposition algorithm.

The resultant approach is an alternative to Euclid's algorithm. The Sylvester resultant and subresultant matrices of two univariate polynomials are discussed in Chapter 5. The degree of the GCD of the exact polynomials $f_1(x)$ and $f_2(x)$ is equal to the rank loss of the resultant matrix, which can be determined by using the SVD.

The coefficients of an AGCD of two inexact polynomials with several close roots are not easily determined using QR decomposition, unlike the case of exact polynomials. A strategy was proposed in [20] to alleviate the instability problem, but it fails when a leading coefficient is lower than 10^{-5}, as shown in [9]. So a new method that uses QR decomposition of the Bezoutian resultant matrix based on pivoting was established to compute the degree of an AGCD [9]. Further, the estimated smallest singular value method was investigated in [92] based on the estimator used in [10].

A method based on optimisation can also be used [17, 20, 45, 46, 47, 62], and the GCD computed by this method satisfies the properties in the definition below.

Definition 3.5. [39]. Consider two polynomials g_1 and g_2 with degrees m and n respectively, and $k < m, n$. Then $d(x)$ is an AGCD of g_1 and g_2 if:

1. $\deg d(x) = k$.

2. $d(x)$ is an exact divisor for the perturbed polynomials $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$.

3. The perturbation norm η is minimised, where

$$\eta = \|g_1(x) - \tilde{g}_1(x)\|^2 + \|g_2(x) - \tilde{g}_2(x)\|^2,$$

The magnitude of the perturbation to be added into an AGCD polynomial was studied by Karmarkar and Lakshman [46]. They describe the characterisation of the minimum perturbation that has to be added to the given two polynomials so that the perturbed polynomials have a non-trivial GCD [46]. Likewise, Chin and Corless [17] solved a non-linear optimisation problem which has a similar formulation of the minimisation problem function in [46], but differs in terms of the function that is minimised. In other words, Karmarkar and Lakshman in [46, 47] addressed the optimisation problem as a function of the common roots of the polynomial that were perturbed, whereas Chin and Corless in [17] addressed it as a function of common divisor coefficients and presumed that the degree of an AGCD is known *a priori*.

In summary, several methods for computing an AGCD have been investigated in the literature. Recent interest has focused on structured matrix methods [45, 58, 86]. Many works based on this approach lies in the fact that an estimation of the noise level must be made. In other words, a threshold is required to specify the minimum perturbation to be added to the inexact polynomials in order to have a non-constant GCD. Some studies have focused on this limitation and have introduced a new strategy based on the data-driven theory which does not require a certain threshold [39, 84, 88, 91]. Mainly, they apply preprocessing operations to the Sylvester resultant matrix of two polynomials whose coefficients

are unknown exactly to obtain an accurate AGCD.

3.4.1 The application of AGCD in image restoration

In image restoration context, Pillai et al. [69] performed AGCD computation to solve the BID problem and described two cases; in one a pair of blurred images of the same scene are given, and in the other a single blurred image is available, such that both cases lead to a solution of the original image. He suggests that the PSF can be obtained using a partition of the distorted image. The choice of any part requires it to contain the whole information of the PSF, and thus he assumed each line of the image is blurred by the linear motion blur. Consequently, the GCD of two part will result in the PSF and then the restoration of the original image [69]. They operate the deblurring process on a quite higher SNR about 45 dB, while the work presented in this thesis can operate on lower SNR. However, the computations in this method are performed in $z-$domain using the DFT, which leads to quantisation errors in the obtained solution [69].

Some other work has used this idea on one degraded image formed as

$$\mathcal{G} = \big((\mathcal{F} + \mathcal{E}) \otimes \mathcal{P}\big) + \mathcal{N} \quad \in \mathbb{R}^{(M+r)\times(N+r)}, \tag{3.5}$$

where $\mathcal{G}, \mathcal{F}, \mathcal{E}, \mathcal{P}$ and $\mathcal{N}$ are the blurred and original images, measurement errors, the PSF, and the additive noise respectively.

This model is used to restore a PSF, that is $\mathcal{P}$, of an equal dimensions $r \times r$, without using the DFT [24]. The method used in [24], was considered only for a small blurring function of order 3×3 pixels, and the deblurring method was applied only on the first and the last rows or/and columns of the blurred image. The use

of DFT in the GCD computation reduces the computational time; however, the DFT is computed approximately which leads into errors in the GCD computation. The proposed work in [24] has determined the PSF from a degraded image using Eq. 3.5, where the measurement error is added to the exact image and not to the PSF. Moreover, the implemented GCD algorithm in this method requires a *priori* threshold to define the noise [24].

The solution of the BID problem in this research differs on the work in [24, 69], that prior knowledge of the PSF and noise are not required, moreover, the computation are performed without using the DFT. The previous method for computing the degree of an AGCD in [39, 84, 88], in power basis is extended and developed, in this research. Under this condition, the BID approach is developed to include two important techniques. Initially, a set of preprocessing operations is applied on the two inexact polynomials, as illustrated in Chapter 5. Secondly, the degree of an AGCD is computed based on two methods used in [84] that are considered in Chapter 6. The coefficients of AGCD polynomial are computed using the nonlinear structure preserving matrix method [87], and least squares method [91].

3.5 Summary

This chapter has defined the GCD and reviewed various definitions of an AGCD. Definition 3.4 of an AGCD is applied in this research. An overview of the AGCD computation approaches was discussed along with some of their limitations. Given these points, the BID problem is addressed in the next chapter using polynomial representation.

Chapter 4

The image deblurring by polynomial computations

An overview of the BID problem was given in Chapter 2. It refers to the process of determining both the exact image and the PSF from an inexact image. The computation of the PSF is connected to the computation of an AGCD of two univariate polynomials. This chapter introduces the BID problem in polynomial form, in order to use an AGCD method. Section 4.1 defines the image blurring components in polynomial form. The problem of BID will be considered for the separable PSF in Section 4.2 and for bivariate case in Section 4.3.

4.1 Deconvolution and polynomial

Chapter 2 described some of the image restoration methods based on the blurring model in Eq. 2.1. This section describes possible forms of the blurring model

that could be applied to an exact image, in polynomial form. Suppose that $\mathcal{F}$, $\mathcal{G}$, P, $\mathcal{E}$, and $\mathcal{N}$ are the exact image, the blurred image, the PSF array, the measurement error, and the noise respectively. Then, the image distortion model can be obtained as follows.

$$\mathcal{G} = \mathcal{F} \otimes (\mathcal{P} + \mathcal{E}) + \mathcal{N}, \tag{4.1}$$

where $\otimes$ denotes the convolution operation.

This model is assumed in this research, where the PSF is not known exactly and uncertainty is represented by $\mathcal{E}$, and additive noise is present. Consider a matrix $F \in \mathbb{R}^{M \times N}$ of an exact image $\mathcal{F}$ given by

$$F = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{0,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix},$$

whose entries are the coefficients of a bivariate polynomial $f(i,j)$, where x and y are pixel locations,

$$\begin{aligned} f(x,y) = \sum_{i=0}^{M-1} \sum_{i=0}^{N-1} a_{ij} x^i y^j \qquad i &= 0, \ldots, M-1, \\ j &= 0, \ldots, N-1. \end{aligned} \tag{4.2}$$

Similarly, the matrix $P \in \mathbb{R}^{C \times D}$ of the PSF array whose entries are the coefficients of a bivariate polynomial $p(x,y)$ is given by

$$\begin{aligned} p(x,y) = \sum_{k=0}^{C-1} \sum_{l=0}^{D-1} p_{kl} x^k y^l, \qquad k &= 0, \ldots, C-1, \\ l &= 0, \ldots, D-1. \end{aligned} \tag{4.3}$$

Obviously, the dimensions of the PSF in the horizontal and vertical directions are not assumed to be equal [38]. A blurred image is the result of blending the

PSF all over the clean image, in both directions, row-wise and column-wise. It follows that the blurred image $\mathcal{G}$ is stored in a matrix $G \in \mathbb{R}^{(M+C-1)\times(N+D-1)}$ whose entries are the coefficients of a bivariate polynomial $g(x, y)$, such that

$$g(x, y) = f(x, y)p(x, y).$$

Eqs. 4.2 and 4.3 yield

$$g(x, y) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{k=0}^{C-1}\sum_{l=0}^{D-1} f(i, j)p(k, l)x^{(i+k)}y^{(j+l)}.$$

The substitutions $s = i + k$ and $t = j + l$ give

$$g(x, y) = \sum_{i,j,s,t} f(i, j)p(s - i, t - j)x^s y^t. \tag{4.4}$$

This equation shows that the pixel values of the blurred image $\mathcal{G}$ are equal to the 2D convolution of the pixel values of the exact image $\mathcal{F}$ and the PSF [38]. The coefficient of $x^s y^t$ in the bivariate polynomial $g(x, y)$ of $\mathcal{G}$ is given by

$$\sum_{i,j,s,t} f(i, j)p(s - i, t - j), \tag{4.5}$$

and this is the value of the pixel at (s, t) in $\mathcal{G}$.

The product in Eq. 4.4 shows that the blurred image $\mathcal{G}$ is represented by a bivariate polynomial of degrees $(M + C - 2)$ and $(N + D - 2)$ in x and y respectively. Thus, its coefficients are stored in a matrix of order $(M + C - 1) \times (N + D - 1)$. This matrix is larger than the matrix F of $\mathcal{F}$ because of the extra rows at the top and bottom and the extra columns on the left and right of G. As was explained in Chapter 2, these extra rows and columns in G define the boundary conditions. Consequently, the bivariate polynomial $g(x, y)$ is of higher degrees than the degrees of the bivariate polynomial $f(x, y)$.

The polynomial representation of Eq. 4.1 is given by

$$g(x, y) = f(x, y)\big(p(x, y) + e(x, y)\big) + n(x, y), \tag{4.6}$$

where g, f, p, e and n are the polynomial representations of the blurred image $\mathcal{G}$, the exact image $\mathcal{F}$, the PSF array $\mathcal{P}$, the measurement errors $\mathcal{E}$, and additive noise $\mathcal{N}$ respectively. The model in Eq. 4.6 is an equation of the following form:

$$g(x,y) = f(x,y)d(x,y) + n(x,y), \tag{4.7}$$

where d is an approximate polynomial of the PSF. If the following is assumed,

$$\| n(x,y)\| \ll \| f(x,y)d(x,y)\|$$

where $\| \cdot \|$, without any subscript, is refer to $2-$norm $\| \cdot \|_2$, defined by

$$\|a\|_2 = \left(|a_1|^2 + |a_2|^2 + \cdots + |a_n|^2\right)^{1/2}, \quad a \in \mathbb{R}^n.$$

Then, Eq. 4.7 can be approximated to

$$g(x,y) \approx f(x,y)d(x,y). \tag{4.8}$$

The proposed solution to the BID problem implements the above equations to reconstruct a PSF. There are two cases in which the BID problem can be solved, including separable and nonseparable. The next section illustrates the BID solution to reconstruct a separable PSF.

4.2 The separable PSF

Section 2.3 has described the convolution of a PSF and an exact image. The blurred image $\mathcal{G}$ is obtained as a result of blending the PSF all over the clean image $\mathcal{F}$, to be more precise, in row-wise or/and column-wise dimensions. It means that each pixel in $\mathcal{G}$ is represented as a multiplication of the PSF and the corresponding pixel in $\mathcal{F}$, and the PSF is spatially invariant. Section 4.2.1

explains the solution of the BID problem for 1D PSF. Section 4.2.2 extends the solution of the BID problem to solve the 2D separable PSF.

4.2.1 One dimensional PSF

This section assumes that the exact image $\mathcal{F}$ is blurred by a 1D PSF across each row or each column, resulting in the equations below:

$$g(x,y) = f(x,y)p_r(y), \tag{4.9}$$

$$g(x,y) = f(x,y)p_c(x). \tag{4.10}$$

where $p_r(y)$ and $p_c(x)$ are polynomials represent the blurring in row-wise and column-wise respectively. If Eq. 4.9 is applied, the problem can be solved using two distinct rows $\mathcal{R}_1$ and $\mathcal{R}_2$ of the blurred image $\mathcal{G}$, such that each contains the full PSF. However, the following theory can be modified when Eq. 4.10 is assumed. The application of Eq. 4.8 to $\mathcal{R}_1$ and $\mathcal{R}_2$, produces

$$h_1(x,y) \approx u_1(x,y)d_{r_1}(y), \qquad \text{and} \qquad h_2(x,y) \approx u_2(x,y)d_{r_2}(y). \tag{4.11}$$

where $h_1(x,y)$ and $h_2(x,y)$ are the polynomials represent $\mathcal{R}_1$ and $\mathcal{R}_2$ respectively. It is clear that the blurring in row-wise $d_{r_1}(y)$ and $d_{r_2}(y)$ are approximately similar:

$$d_{r_1}(y) = p_r(y) + e_1(y), \qquad \text{and} \qquad d_{r_2}(y) = p_r(y) + e_2(y),$$

and they only vary in their measurement errors. These errors are considered to be small, such that $d_{r_1}(y) \approx d_{r_2}(y)$, and therefore they substituted by $d_r(y)$. It follows that Eqs. 4.11 become

$$h_1(x,y) \approx u_1(x,y)d_r(y), \qquad \text{and} \qquad h_2(x,y) \approx u_2(x,y)d_r(y). \tag{4.12}$$

Conceding that $u_1(x, y)$ and $u_2(x, y)$ are two distinct regions of the exact image $\mathcal{F}$, such that they are coprime, and therefore an AGCD computation of polynomials $h_1(x, y)$ and $h_2(x, y)$ is required to reconstruct an approximation of the 1D PSF,

$$d_r(y) = \text{AGCD}\left(h_1(x, y), h_2(x, y)\right).$$

Once the 1D PSF is estimated, the restored image can be obtained using polynomial division

$$\tilde{f}(x, y) = g(x, y)/d_r(y).$$

where $\tilde{f}(x, y)$ is a bivariate polynomial represents the restored image $\tilde{\mathcal{F}}$. The column-wise blurring can be calculated using two columns of $\mathcal{G}$, and the AGCD polynomial is expressed as $d_c(x)$.

The next section presents the theory when 2D blurring is applied on an image.

4.2.2 Two dimensional PSF

It was shown in Eq. 2.12 that a 2D PSF can be decomposed into a column-vector and a row-vector if it has rank one. On the assumption of 2D separable PSF, the following holds:

$$p(x, y) = p_c(x)p_r(y), \tag{4.13}$$

Introducing the measurement errors to Eq. 4.13 yields the following approximation of the separable PSF

$$d(x, y) = d_c(x)d_r(y), \tag{4.14}$$

It follows that a polynomial $g(x, y)$ of image $\mathcal{G}$ can be written as

$$g(x, y) \approx d_c(x)f(x, y)d_r(y), \quad \text{or} \quad g(x, y) \approx d_r(y)f(x, y)d_c(x). \tag{4.15}$$

Following the description in Section 2.5.2, it is appropriate to follow one form of $g(x, y)$ since the same steps can be applied to the other. Given two distinct rows $\mathcal{R}_1$ and $\mathcal{R}_2$, and two distinct columns $\mathcal{R}_3$ and $\mathcal{R}_4$ of the blurred image $\mathcal{G}$, such that $\mathcal{R}_1$ and $\mathcal{R}_2$ contain the row-wise components of the PSF and $\mathcal{R}_3$ and $\mathcal{R}_4$ contain the column-wise components of the PSF. The application of Eq. 4.15 to $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_4$, produces

$$\begin{aligned} r_k(x, y) &\approx u_k(x, y)d_r(y), \quad k = 1, 2, \\ c_k(x, y) &\approx u_k(x, y)d_c(x), \quad k = 1, 2, \end{aligned} \tag{4.16}$$

where $r_k(x, y)$ for $k = 1, 2$ are the polynomials represent $\mathcal{R}_1$ and $\mathcal{R}_2$, and $c_k(x, y)$ for $k = 1, 2$ are the polynomials represent $\mathcal{R}_3$ and $\mathcal{R}_4$.

The separable PSF can be calculated using two AGCD computations, one considers rows $r_k(x, y)$, and the other considers columns $c_k(x, y)$ for $k = 1, 2$, resulting in AGCD polynomials $d_r(y)$ and $d_c(x)$. In particular,

$$d_r(y) = \text{AGCD}\big(r_1(x, y), r_2(x, y)\big), \qquad \text{and} \qquad d_c(x) = \text{AGCD}\big(c_1(x, y), c_2(x, y)\big).$$

This method is followed by another operation to restore the original image $\mathcal{F}$ from the degraded version $\mathcal{G}$ using polynomial division and Eq. 4.14.

$$\tilde{f}(x, y) = g(x, y)/d(x, y).$$

The PSF is identified using AGCD computation, based on the work done in [87, 88] that will be discussed in Chapter 6. While, the method used to restore the exact image will be discussed in Chapter 8.

Example 4.1. Suppose that an exact image $\mathcal{F}$ of size 5×5 is represented by matrix F, and a separable Gaussian blur of size 3×3 is represented by matrix P

as below:

$$F = \begin{bmatrix} 162 & 143 & 165 & 124 & 162 \\ 184 & 143 & 165 & 184 & 162 \\ 124 & 165 & 184 & 143 & 124 \\ 162 & 124 & 162 & 143 & 184 \\ 184 & 143 & 165 & 124 & 162 \end{bmatrix}, \quad \text{and} \quad P = \begin{bmatrix} 0.0277 & 0.1110 & 0.0277 \\ 0.1110 & 0.4452 & 0.1110 \\ 0.0277 & 0.1110 & 0.0277 \end{bmatrix}.$$

Matrix P is of rank one, and can be expressed as the product of its decomposed a column-vector and a row-vector as:

$$P = \begin{bmatrix} 0.1664 \\ 0.6672 \\ 0.1664 \end{bmatrix} \begin{bmatrix} 0.1664 & 0.6672 & 0.1664 \end{bmatrix}.$$

Assuming the zero boundary condition, the convolution of $\mathcal{F}$ with PSF yields an image $\mathcal{G}$ of size 7×7 denoted by matrix G:

$$G = \begin{bmatrix} 4.4846 & 21.9429 & 24.9271 & 25.70850 & 22.8178 & 21.4170 & 4.4845 \\ 23.0779 & 112.3849 & 125.5034 & 130.4706 & 120.9870 & 108.9682 & 22.4689 \\ 28.3437 & 138.0706 & 154.1802 & 164.4233 & 165.4400 & 131.6927 & 25.9014 \\ 23.3438 & 119.3261 & 155.9236 & 168.8567 & 152.7898 & 118.5447 & 23.3438 \\ 26.5104 & 128.6090 & 143.5548 & 156.4268 & 149.2956 & 136.9357 & 28.3437 \\ 24.9112 & 119.2113 & 125.1444 & 128.4766 & 116.9613 & 110.2756 & 23.0779 \\ 5.0935 & 24.3852 & 25.5361 & 25.7085 & 22.81780 & 21.4170 & 4.4845 \end{bmatrix}.$$

Each row of F represents a univariate polynomial of degree 4, likewise for each column in F. Matrix P is decomposed into a row and a column, such that each is represented by a univariate polynomial, both of degree 2. Therefore the multiplication of each row in F with the decomposed row vector of P yields a univariate polynomial of degree 6. The same applies for each column in F with the decomposed column vector of P. It is easily seen that due to the assumption

of the zero boundary condition, the extra rows and columns at the border of G are smaller in value than the other elements of G. The 2D representation G of $\mathcal{G}$ has the same size as F of $\mathcal{F}$, if the extra rows and columns at the border of G are omitted. Accordingly, the computation of the GCD of two univariate polynomials can be performed using two distinct rows of G or/and two distinct columns of G to recover the PSF.

For example, let R_1 and R_2 be two random rows of G respectively, then their univariate polynomial representations are:

$$\begin{aligned} r_1(y) &= (4.4845)y^6 + (21.4170)y^5 + (22.8178)y^4 + (25.7085)y^3 + (24.9270)y^2 \\ &+ (21.9429)y + (4.4845), \\ r_2(y) &= (4.4845)y^6 + (21.4170)y^5 + (22.8178)y^4 + (25.7085)y^3 + (25.5361)y^2 \\ &+ (24.3852)y + (5.0935). \end{aligned}$$

To compute the horizontal components of the PSF in the noiseless condition, a calculation of GCD must be performed, such that

$$\begin{bmatrix} 0.1664 & 0.6672 & 0.1664 \end{bmatrix} = \mathrm{GCD}\left(r_1(y), r_2(y)\right).$$

The same result can be achieved for two distinct columns.

If the model in Eq. 4.1 has a measurement error 10^{-3} and an additive noise of 10^{-4}, then by assuming the zero boundary condition the blurred image $\mathcal{G}$ is given

by matrix G as follows:

$$G = \begin{bmatrix} 4.4885 & 21.9635 & 24.9489 & 25.7309 & 22.8360 & 21.43474 & 4.4860 \\ 23.0987 & 112.4771 & 125.5983 & 130.5606 & 121.0833 & 109.0446 & 22.4814 \\ 28.3684 & 138.16629 & 154.2882 & 164.5306 & 165.5437 & 131.7739 & 25.9169 \\ 23.3628 & 119.4108 & 156.0326 & 168.9605 & 152.8872 & 118.6223 & 23.3573 \\ 26.5326 & 128.6993 & 143.6468 & 156.5233 & 149.3942 & 137.0284 & 28.3609 \\ 24.9323 & 119.2801 & 125.2269 & 128.5603 & 117.0380 & 110.3429 & 23.0948 \\ 5.0942 & 24.3898 & 25.5435 & 25.7163 & 22.8259 & 21.4226 & 4.4889 \end{bmatrix}.$$

The horizontal components of the PSF can be computed by choosing r_1 and r_2, which are two univariate polynomials of two rows of $\mathcal{G}$:

$$\begin{aligned} r_1(y) &= (4.4860)y^6 + (21.4347)y^5 + (22.8360)y^4 + (25.7309)y^3 + (24.9489)y^2 \\ &+ (21.9635)y + (4.4885), \\ r_2(y) &= (4.4889)y^6 + (21.4226)y^5 + (22.8259)y^4 + (25.7163)y^3 + (25.5435)y^2 \\ &+ (24.3898)y + (5.0942). \end{aligned}$$

The GCD computation here reduces to an AGCD computation:

$$d_r(y) = \text{AGCD}\big(r_1(y), r_2(y)\big).$$

Likewise if $c_1(x)$ and $c_2(x)$ are two univariate polynomials corresponding to two random columns of G respectively, then

$$d_c(x) = \text{AGCD}\big(c_1(x), c_2(x)\big).$$

If two blurred images of different scenes are given, such that each contains an equal separable PSF, eg. taken by the same device, then a row and a column from each image are chosen in order to perform two AGCD computations.
This is not applicable when the exact images are convolved with a PSF that is not separable, and thus a bivariate problem is introduced. The next section

will examine the polynomial form of bivariate problem, when the PSF is not separable.

4.3 The nonseparable PSF

This section presents the BID problem in its bivariate polynomial form, in particular, it considers the PSF that has a rank larger than one. This is referred to as a nonseparable PSF, where the PSF array that is given by matrix P cannot be decomposed into a column vector and a row vector.

Before proceeding with this problem, it is important to define an ordering system for an image. The entries of a matrix representation of an image with dimensions $M \times N$ is represented in bivariate power basis as follows:

$$S = \begin{bmatrix} 1 & y & y^2 & \dots & y^{N-1} \\ x & xy & xy^2 & \dots & xy^{N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x^{M-1} & x^{M-1}y & x^{M-1}y^2 & \dots & x^{M-1}y^{N-1} \end{bmatrix}. \tag{4.17}$$

Matrix S stores the coefficients of a bivariate polynomial $s(x, y)$, that contains $x^{i-1}y^{j-1}$, for $i = 1, \cdots, M$ and $j = 1, \cdots, N$.An alternative ordering system must be arranged for the coefficients of the corresponding bivariate polynomial, so that it can be represented in an equivalent univariate polynomial. Thus, the representation in S of a matrix is transformed into univariate power basis as:

$$T = \begin{bmatrix} 1 & z^M & z^{2M} & \dots & z^{M(N-1)} \\ z & z^{M+1} & z^{2M+1} & \dots & z^{M(N-1)+1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ z^{M-1} & z^{2M-1} & z^{3M-1} & \dots & z^{MN-1} \end{bmatrix}. \tag{4.18}$$

Matrix T stores the coefficients of a univariate polynomial $t(z)$, which contains z^{k-1} for $k = 1, \cdots, MN$.

The relation between the index (i, j) in matrix S and the index $(k-1)$ in matrix T must be defined. This relation can be derived using the division and modulo operators, such that

$$k = i + M(j-1), \tag{4.19}$$

and

$$i = \big((k-1) \text{ MOD } M\big) + 1, \quad j = \big((k-1) \text{ DIV } M\big) + 1, \tag{4.20}$$

where DIV and MOD denote the division and modulo operators respectively, for $i = 1, \cdots, M; j = 1, \cdots, N$; $k = 1, \cdots, MN$.

, For example, the equivalent univariate polynomial of a bivariate polynomial $f(x, y)$ using univariate power basis is $f(z)$ of degree $(MN - 1)$, such that its coefficients are arranged in a matrix $F \in \mathbb{R}^{M \times N}$.

Now consider the blurred versions of the exact images $\mathcal{F}_1$ and $\mathcal{F}_2$ expressed in polynomial form as:

$$\begin{aligned} g_1(x, y) &= f_1(x, y)\big(p(x, y) + e_1(x, y)\big) + n_1(x, y), \\ g_2(x, y) &= f_2(x, y)\big(p(x, y) + e_2(x, y)\big) + n_2(x, y), \end{aligned}$$

where e denotes the measurement errors and n denotes the additive noises.

The PSF is assumed to be spatially invariant, and thus the PSF is assumed to be equally assigned in each pixel of $\mathcal{G}_1$ and $\mathcal{G}_2$. If $(e_1 = e_2 = 0)$ and $(n_1 = n_2 = 0)$, then GCD computations of $g_1(x, y)$ and $g_2(x, y)$ in x and y variables lead to the exact PSF, p. However, if $(e_1, e_2, n_1, n_2 \neq 0)$, then an approximation of the common factor p is estimated. Using Eq. 4.8 in the presence of noise, the

polynomial forms of $\mathcal{G}_1$ and $\mathcal{G}_2$ are approximated into the following:

$$g_1(x,y) \approx f_1(x,y)d(x,y), \qquad \text{and} \qquad g_2(x,y) \approx f_2(x,y)d(x,y), \tag{4.21}$$

where

$$d(x,y) \approx d_1(x,y) \approx d_2(x,y).$$

It is assumed that $\mathcal{F}_1$ and $\mathcal{F}_2$ are dissimilar. Therefore, their polynomial representations are considered to be coprime. It follows that

$$d(x,y) = \text{AGCD}\big(g_1(x,y), g_2(x,y)\big).$$

The corresponding matrices G_1 and G_2 of $g_1(x,y)$ and $g_2(x,y)$, are represented as Eq. 4.17. Alternative representations of G_1 and G_2 are assumed, so each can be expressed in z-basis as shown in Eq. 4.18. It follows that the matrices G_1 and G_2 in univariate power basis are transformed into column vector representation by stacking their columns. Accordingly, Eq. 4.21 is written as

$$g_1(z) \approx f_1(z)d(z), \qquad \text{and} \qquad g_2(z) \approx f_2(z)d(z). \tag{4.22}$$

Then

$$d(z) = \text{AGCD}\big(g_1(z), g_2(z)\big).$$

The recovered PSF is derived by converting the computed AGCD from a univariate polynomial $d(z)$ to a bivariate polynomial $d(x,y)$, using Eqs. 4.20. Then the exact images $\mathcal{F}_1$ and $\mathcal{F}_2$ are reconstructed using polynomial division:

$$\tilde{f}_k(x,y) = g_k(x,y)/d(x,y), \quad k = 1,2. \tag{4.23}$$

Example 4.2. Consider matrices F, P and G given by

$$F = \begin{bmatrix} 0 & 3 & 1 \\ -1 & 1 & -4 \\ 2 & 5 & 3 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 3 & 7 & 5 & 1 \\ -1 & 2 & -2 & -7 & -4 \\ 1 & 10 & 17 & 16 & 4 \\ 2 & 3 & 4 & -7 & -4 \\ 0 & 4 & 12 & 11 & 3 \end{bmatrix},$$

where matrix G represents the convolution of F with P. The entries of matrix G are stored in bivariate power basis as follows:

$$G = \begin{bmatrix} 1 & y & y^2 & y^3 & y^4 \\ x & xy & xy^2 & xy^3 & xy^4 \\ x^2 & x^2y & x^2y^2 & x^2y^3 & x^2y^4 \\ x^3 & x^3y & x^3y^2 & x^3y^3 & x^3y^4 \\ x^4 & x^4y & x^4y^2 & x^4y^3 & x^4y^4 \end{bmatrix}. \tag{4.24}$$

The matrices F and P are expressed in the bivariate polynomial form as:

$$\begin{aligned} f(x,y) &= 3x^2y^2 + 5x^2y + 2x^2 - 4xy^2 + xy - x + y^2 + 3y, \\ p(x,y) &= 1 + x + 2y + y^2 + x^2y^2. \end{aligned}$$

The product of $f(x,y)$ and $p(x,y)$ yields a bivariate polynomial $g(x,y)$ which corresponds to the matrix G, such that

$$\begin{aligned} g(x,y) &= 3x^4y^4 + 11x^4y^3 + 12x^4y^2 + 4x^4y - 4x^3y^4 - 7x^3y^3 + 4x^3y^2 + 3x^3y \\ &+ 3x^3 + 4x^2y^4 + 16x^2y^3 + 17x^2y^2 + 10x^2y + x^2 - 4xy^4 - 7xy^3 - 2xy^2 \\ &+ 2xy - x + y^4 + 5y^3 + 7y^2 + 3y. \end{aligned}$$

The bivariate power basis of matrix G in Eq. 4.24 is transformed into univariate

power basis, such that its entries are stored in G as shown below:

$$G = \begin{bmatrix} 1 & z^5 & z^{10} & z^{15} & z^{20} \\ z & z^6 & z^{11} & z^{16} & z^{21} \\ z^2 & z^7 & z^{12} & z^{17} & z^{22} \\ z^3 & z^8 & z^{13} & z^{18} & z^{23} \\ z^4 & z^9 & z^{14} & z^{19} & z^{24} \end{bmatrix}.$$

The bivariate problem can be transformed then into univariate problems, such that $f(x, y)$ $p(x, y)$ and $g(x, y)$ become

$$\begin{aligned} f(z) &= 3z^{12} - 4z^{11} + z^{10} + 5z^7 + z^6 + 3z^5 + 2z^2 - z, \\ p(z) &= z^{12} + z^{10} + 2z^7 + 2z^5 + z^2 + z + 1, \\ g(z) &= 3z^{24} - 4z^{23} + 4z^{22} - 4z^{21} + z^{20} + 11z^{19} - 7z^{18} + 16z^{17} - 7z^{16} + 5z^{15} \\ &+ 12z^{14} + 4z^{13} + 17z^{12} - 2z^{11} + 7z^{10} + 4z^9 + 3z^8 + 10z^7 + 2z^6 + 3z^5 \\ &+ 2z^3 + z^2 - z + 1. \end{aligned}$$

It can be clearly seen that a blurred image can be rearranged as a column vector that represents a univariate polynomial. Therefore an AGCD computation [87] using a Sylvester matrix of two univariate polynomials is applicable in this case. However, a Sylvester matrix is assumed to be large for bivariate polynomials even with a reasonable order of x and y. Therefore, it is necessary to reduce the degree of the bivariate polynomials in order to perform the computations.
The method described in this section to simplify the problem to univariate polynomial is still considered non-effective for larger degree.
In this research, the 2D problem is simplified into two 1D problems, by using two AGCD computations based on the Sylvester resultant matrix to estimate the separable PSF. Moreover, high performance computing (HPC) is implemented

to estimate the nonseparable PSF, as it will be shown in Chapter 7. Before proceeding with the blind PSF estimation, it is important to introduce the Sylvester resultant matrix and its subresultant matrices, that is needed to calculate the GCD and an AGCD of a pair of polynomials. The relation between the Sylvester matrix type and the GCD computation is explained in the next chapter.

4.4 Summary

The polynomial representation of the BID problem, including the univariate and bivariate polynomials, was introduced in this chapter. Two methods to calculate the PSF have been outlined. The first method aims to restore separable types of the PSF and the second method aims to restore general types of the PSF; more precisely, separable and nonseparable. These methods require the calculation of an AGCD of two univariate polynomials, whose coefficients are unknown exactly, using a Sylvester resultant matrix. What follows is an overview of the Sylvester resultant matrix type and its relation to the GCD computations.

Chapter 5

Sylvester resultant matrix and its modfied form

5.1 Introduction

A new solution of the BID problem using polynomial computations has been described in Chapter 4. This solution uses the Sylvester resultant matrix to estimate the PSF. The Sylvester resultant and its k subresultant matrices, denoted by $S(f_1(x), f_2(x))$ and $S_k(f_1(x), f_2(x))$ respectively, can be used for the computation of the GCD of two univariate polynomials $f_1(x)$ and $f_1(x)$ whose coefficients are known exactly. In particular, $f_1(x)$ and $f_2(x)$ have a non-constant GCD if their Sylvester matrix is singular [4]. In this case, the degree of the GCD of $f_1(x)$ and $f_2(x)$ is equal to the rank loss of their Sylvester matrix. The coefficients of the GCD can be retrieved by reducing the Sylvester resultant matrix to an upper triangular structure form using the decompositions LU or QR and the coefficients of GCD will be found in the last non zero row of U or R. The polynomials are

co-prime if their Sylvester matrix is non-singular (i.e the determinant of Sylvester matrix is not equal to zero).

This chapter will examine the Sylvester resultant type in power basis and its properties, for its application to GCD computations, using the theory proposed in [84]. Section 5.2 describes the Sylvester resultant matrix, and its subresultant matrices will be described in Section 5.3. A modification to the Sylvester matrix and its subresultant matrices of two polynomials $g_1(x)$ and $g_2(x)$ whose coefficients are perturbed by noise, denoted by $S_k(g_1(x), g_2(x))$, will be introduced in Section 5.4.

5.2 Sylvester resultant matrix

As already noted, a sufficient and necessary condition for two exact polynomials to have a non-constant GCD is that the determinant of their Sylvester matrix $S(f_1(x), f_2(x))$ equals zero. The Sylvester matrix in [4, 23] can be illustrated by the following definition.

Definition 5.1. [84]. Let $f_1(x)$ and $f_2(x)$ be two exact polynomials of positive degree, as in the following:

$$f_1(x) = \sum_{i=0}^{m} \hat{a}_{m-i} x^{m-i}, \qquad i = 0, \ldots, m, \qquad \hat{a}_m \neq 0. \tag{5.1}$$

$$f_2(x) = \sum_{j=0}^{n} \hat{b}_{n-j} x^{n-j}, \qquad j = 0, \ldots, n, \qquad \hat{b}_n \neq 0. \tag{5.2}$$

Then the Sylvester matrix of two polynomials $f_1(x)$ and $f_2(x)$ is $S(f_1, f_2) \in \mathbb{R}^{(m+n)\times(m+n)}$,

$$S(f_1,f_2)=\left[\begin{array}{cccc|cccc} \hat{a}_m & & & & \hat{b}_n & & & \\ \hat{a}_{m-1} & \hat{a}_m & & & \hat{b}_{n-1} & \hat{b}_n & & \\ \vdots & \hat{a}_{m-1} & \ddots & & \vdots & \hat{b}_{n-1} & \ddots & \\ \hat{a}_1 & \vdots & \ddots & \hat{a}_m & \hat{b}_1 & \vdots & \ddots & \hat{b}_n \\ \hat{a}_0 & \hat{a}_1 & \ddots & \hat{a}_{m-1} & \hat{b}_0 & \hat{b}_1 & \ddots & \hat{b}_{n-1} \\ & \hat{a}_0 & \ddots & \vdots & & \hat{b}_0 & \ddots & \vdots \\ & & \ddots & \hat{a}_1 & & & \ddots & \hat{b}_1 \\ & & & \hat{a}_0 & & & & \hat{b}_0 \end{array}\right],$$

where the coefficients $\hat{a}_i$ of the polynomial $f_1(x)$ occupy the first n columns of $S(f_1, f_2)$ and the coefficients $\hat{b}_i$ of the polynomial $f_2(x)$ occupy the last m columns of $S(f_1, f_2)$, [84]. The resultant of $f_1(x)$ and $f_2(x)$ is denoted by $\mathrm{Res}(f_1, f_2)$ such that

$$Res(f_1(x), f_2(x)) = \det(S(f_1(x), f_2(x))).$$

The polynomials $f_1(x)$ and $f_2(x)$ have a common divisor if an only if

$$\mathrm{Res}(f_1(x), f_2(x)) = 0,$$

in which case, there exists a common divisor polynomial $\hat{d}_k(x), k = 1, \ldots, \hat{d}$ that satisfies

$$\hat{d}_k(x) = \frac{f_1(x)}{\hat{u}_k(x)} = \frac{f_2(x)}{\hat{v}_k(x)}, \qquad \text{and} \qquad \hat{d} = \deg \mathrm{GCD}(f_1(x), f_2(x)), \tag{5.3}$$

[84], where $\hat{u}_k(x)$ and $\hat{v}_k(x)$ are quotient polynomials of degree $\hat{u}_k < m$ and $\hat{v}_k < n$ respectively:

$$\hat{u}_k(x) = \sum_{i=0}^{m-k} \hat{u}_{k,m-i} x^{m-i}, \qquad \hat{v}_k(x) = \sum_{j=0}^{n-k} \hat{v}_{k,n-j} x^{n-j}, \tag{5.4}$$

and

$$\hat{d}_k(x) = \sum_{i=0}^{k} \hat{d}_{k,k-i} x^{k-i}.$$

Eq. 5.3 can be represented as

$$\hat{v}_k(x) f_1(x) = \hat{u}_k(x) f_2(x), \quad k = 1, \ldots, \hat{d}, \tag{5.5}$$

where $\hat{u}_k(x)$ and $\hat{v}_k(x)$ are equal to the zero polynomial for $k = \hat{d}+1, \ldots, \min(m, n)$, because $\deg \mathrm{GCD}(f_1(x), f_2(x)) = \hat{d}$, that is:

$$\hat{u}_k(x) \equiv 0, \quad \hat{u}_k(x) \equiv 0, \quad k = \hat{d} + 1, \ldots, \min(m, n). \tag{5.6}$$

Accordingly, Eq. 5.5 and Eq. 5.6 can be written in matrix form as

$$[C_k \;\; D_k] \begin{bmatrix} \hat{v}_k \\ -\hat{u}_k \end{bmatrix} = S_k \begin{bmatrix} \hat{v}_k \\ -\hat{u}_k \end{bmatrix} = 0, \quad k = 1, \ldots, \min(m, n), \tag{5.7}$$

where the Sylvester matrix can be expressed as two Toeplitz matrices

$C_k = C_k(f_1) \in \mathbb{R}^{(m+n-k+1)(n-k+1)}$, and $D_k = D_k(f_2) \in \mathbb{R}^{(m+n-k+1)(m-k+1)}$:

$$C_k(f_1) = \begin{bmatrix} \hat{a}_m & & \\ \hat{a}_{m-1} & \ddots & \\ \vdots & \ddots & \hat{a}_m \\ \hat{a}_0 & \vdots & \hat{a}_{m-1} \\ & \ddots & \vdots \\ & & \hat{a}_0 \end{bmatrix}, \quad D_k(f_2) = \begin{bmatrix} \hat{b}_n & & \\ \hat{b}_{n-1} & \ddots & \\ \vdots & \ddots & \hat{b}_n \\ \hat{b}_0 & \vdots & \hat{b}_{n-1} \\ & \ddots & \vdots \\ & & \hat{b}_0 \end{bmatrix},$$

and $S_k = S_k(f_1, f_2)$,

$$\begin{aligned} \hat{u}_k &= \begin{bmatrix} u_{k,0} & u_{k,1} & \ldots & u_{k,m-k+1} & u_{k,m-k} \end{bmatrix} \in \mathbb{R}^{m-k+1}, \\ \hat{v}_k &= \begin{bmatrix} v_{k,0} & v_{k,1} & \ldots & v_{k,n-k+1} & v_{k,n-k} \end{bmatrix} \in \mathbb{R}^{n-k+1}, \end{aligned}$$

such that

$$\begin{aligned} \hat{u}_k, \hat{v}_k &\neq 0, \quad \text{for} \quad k = 1, \ldots, \hat{d}, \\ \hat{v}_k, \hat{v}_k &= 0, \quad \text{for} \quad k = \hat{d}+1, \ldots, \min(m, n), \end{aligned}$$

[84]. It follows that

$$S_k = S_k(f_1, f_2) = [C_k \;\; D_k] \in \mathbb{R}^{(m+n-k+1)(m+n-2k+2)}.$$

The subresultant matrices are determined by deleting some rows and columns from $S(f_1, f_2)$. The kth subresultant matrix $S_k(f_1, f_2) \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+2)}$, where $1 \leq k \leq min(m, n)$, is obtained by removing the last $k-1$ rows of $S(f_1, f_2)$, the last $k-1$ columns of $C_k(f_1)$, and the last $k-1$ columns of $D_k(f_2)$ [84]. The Sylvester subresultant matrix for $k = 1$ reduces to Sylvester resultant matrix $S(f_1, f_2)$, [88], that is:

$$S_1(f_1, f_2) = S(f_1, f_2).$$

The next section describes the relation between the computation of the GCD of $f_1(x)$ and $f_2(x)$ and their Sylvester subresultant matrices $S_k(f_1(x), f_2(x))$.

5.3 Sylvester subresultant matrix

The computation of the GCD of the two exact polynomials $f_1(x)$ and $f_2(x)$ is performed using subresultant matrices. To illustrate how the order of the Sylvester matrix $S(f_1(x), f_2(x))$ and its subresultant matrices $S_k(f_1(x), f_2(x))$ are related to the common divisor of two polynomials $f_1(x)$ and $f_2(x)$, the following theorems are established.

Theorem 5.1. *See also [4, 23, 39].Let $f_1(x)$ and $f_2(x)$ be two polynomials as described in Definition 5.1 and $\hat{d}(x)$ a polynomial of degree k. Let $\hat{u}(x)$ and $\hat{v}(x)$ be two quotient polynomials defined in Eq. 5.4 with degree of $m-k$ and $n-k$ respectively, such that:*

$$\begin{aligned} f_1(x) &= \hat{u}_k(x) \cdot \hat{d}_k(x), \\ f_2(x) &= \hat{v}_k(x) \cdot \hat{d}_k(x). \end{aligned}$$

Then $\hat{d}(x)$ is a common divisor of $f_1(x)$ and $f_2(x)$ if and only if

$$f_1(x) \cdot \hat{v}_k(x) = f_2(x) \cdot \hat{u}_k(x), \qquad k = 1, \ldots, \hat{d}.$$

The degree of the GCD and its coefficients are obtained using Theorems 5.2, and 5.3.

Theorem 5.2. *See also [84, 87]. Let a pair of polynomials $f_1(x)$ and $f_2(x)$ as defined in Eq. 5.1 and Eq. 5.2 respectively have a non-constant GCD of degree $\hat{d}$ and let $S(f_1, f_2)$ be the Sylvester matrix. Then:*

- *the degree $\hat{d}$ of their GCD is equal to the rank loss of the Sylvester resultant matrix $S(f_1, f_2)$, so:*

$$\deg(GCD(f_1, f_2)) = m + n - rank\, S(f_1, f_2).$$

- *The coefficients of the GCD of f_1 and f_2 are in the last non-zero row of the upper triangular forms U and R in the LU and QR decompositions of $S(f_1, f_2)$.*

Theorem 5.3. *See also [4, 23, 39]. The polynomials $f_1(x)$ and $f_2(x)$,as described in 5.1, have a common divisor of degree $k \geq 1$ if the rank of $S_k(f_1, f_2) < (m + n - 2k + 2)$.*

Since $f_1(x)$ and $f_2(x)$ have common divisors of degree $k = 1, \ldots, \hat{d}$, but they do not have common divisors of degree $k = \hat{d}+1, \ldots, \min(m,n)$, it follows that

$$\begin{aligned} \operatorname{rank} S_k(f_1(x), f_2(x)) &< m+n-2k+2, \quad k = 1, \ldots, \hat{d}, \\ \operatorname{rank} S_k(f_1(x), f_2(x)) &= m+n-2k+2, \quad k = \hat{d}+1, \ldots, \min(m,n). \end{aligned}$$

Consequently, the degree $\hat{d}$ of the GCD of $f_1(x)$ and $f_2(x)$ equals the index k of the last subresultant matrix of order $S_1(f_1, f_2), S_2(f_1, f_2), \ldots, S_k(f_1, f_2)$, for which $S_k(f_1, f_2)$ is rank deficient [84]. More precisely, the homogeneous Eq. 5.7 has a non-zero solution for $k = 1, \ldots, \hat{d}$ as shown in Eq. 5.8, and because it is assumed that $f_1(x)$ and $f_2(x)$ have a common divisor of degree k, this implies $\hat{d}_{k,0} \neq 0$, [88]; therefore $\hat{u}_{k,0}, \hat{u}_{k,0} \neq 0$ [84].

It follows that Eq. 5.7 can be transformed from a homogeneous equation to a linear algebraic equation if

$$S_k(f_1, f_2) = [\mathbf{c}_k \;\; A_k], \tag{5.8}$$

where $A_k = A_k(f_1, f_2) \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+1)}$ is the matrix $S_k(f_1, f_2)$ after removing a column, which forms then the vector $\mathbf{c}_k$ [21, 31, 84]. It follows that

$$\begin{aligned} A_k \mathbf{x}_k &= \mathbf{c}_k, \quad k = 1, \ldots, \hat{d}, \\ A_k \mathbf{x}_k &\neq \mathbf{c}_k, \quad k = \hat{d}+1, \ldots, \min(m,n), \end{aligned} \tag{5.9}$$

where

$$\mathbf{x}_k = [v_{k,1} \ldots v_{k,n-k} - u_{k,0} - u_{k,m-k}]^T \in \mathbb{R}^{(m+n-2k+1)}.$$

This leads to Theorem 5.4.

Theorem 5.4. *See also [4, 23, 39]. Consider the polynomials $f_1(x)$ and $f_2(x)$, as described in 5.1, and $k \leq \min(m,n)$ a positive integer. Then the nullity of*

$S_k(f_1, f_2)$ is greater or equal to 1 *if and only if,*

$$A_k \boldsymbol{x}_k = \boldsymbol{c}_k, \quad \boldsymbol{x} \in \mathbb{R}^{m+n-2k+1}, \tag{5.10}$$

has a solution.

Theorems 5.3 and 5.4 yield the following theorem.

Theorem 5.5. *See also [39]. The polynomials $f_1(x)$ and $f_2(x)$ have a common divisor of degree k if and only if Eq. 5.10 possesses a solution.*

It can be shown that Eq. 5.9 possesses a solution for $k = 1, \ldots, \hat{d}$, and therefore c_k lies in the column space of A_k for these values of k and does not possesses a solution for $k = \hat{d}+1, \ldots, \min(m, n)$ [84]. Thus the degree $\hat{d}$ of the GCD of $f_1(x)$ and $f_2(x)$ is equal to the largest values of k for which Eq. 5.9 possesses a solution [84].

The next example describes the construction of Sylvester subresultant matrices.

Example 5.1. Let the polynomial $f_1(x)$ be of degree $m = 6$ and the polynomial $f_2(x)$ be of degree $n = 4$. Then:

$$S_1 = \begin{bmatrix}
\hat{a}_6 & & & & \hat{a}_4 & & & & & \\
\hat{a}_5 & \hat{a}_6 & & & \hat{a}_3 & \hat{a}_4 & & & & \\
\hat{a}_4 & \hat{a}_5 & \hat{a}_6 & & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & & & \\
\hat{a}_3 & \hat{a}_4 & \hat{a}_5 & \hat{a}_6 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & & \\
\hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \hat{a}_5 & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \\
\hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 \\
\hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & & & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \\
 & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & & & & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 \\
 & & \hat{a}_0 & \hat{a}_1 & & & & & \hat{a}_0 & \hat{a}_1 \\
 & & & \hat{a}_0 & & & & & & \hat{a}_0
\end{bmatrix}, \quad S_2 = \begin{bmatrix}
\hat{a}_6 & & \hat{a}_4 & & & & \\
\hat{a}_5 & \hat{a}_6 & \hat{a}_3 & \hat{a}_4 & & & \\
\hat{a}_4 & \hat{a}_5 & \hat{a}_6 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \\
\hat{a}_3 & \hat{a}_4 & \hat{a}_5 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 \\
\hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \\
\hat{a}_1 & \hat{a}_2 & \hat{a}_3 & & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 \\
\hat{a}_0 & \hat{a}_1 & \hat{a}_2 & & & \hat{a}_0 & \hat{a}_1 \\
 & \hat{a}_0 & \hat{a}_1 & & & & \hat{a}_0 \\
 & & \hat{a}_0 & & & &
\end{bmatrix},$$

$$S_3 = \begin{bmatrix} \hat{a}_6 & & \hat{a}_4 & & & \\ \hat{a}_5 & \hat{a}_6 & \hat{a}_3 & \hat{a}_4 & & \\ \hat{a}_4 & \hat{a}_5 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \\ \hat{a}_3 & \hat{a}_4 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 \\ \hat{a}_2 & \hat{a}_3 & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \\ \hat{a}_1 & \hat{a}_2 & & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 \\ \hat{a}_0 & \hat{a}_1 & & & \hat{a}_0 & \hat{a}_1 \\ & \hat{a}_0 & & & & \hat{a}_0 \end{bmatrix}, \quad S_4 = \begin{bmatrix} \hat{a}_6 & \hat{a}_4 & & \\ \hat{a}_5 & \hat{a}_3 & \hat{a}_4 & \\ \hat{a}_4 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 \\ \hat{a}_3 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \\ \hat{a}_2 & \hat{a}_0 & \hat{a}_1 & \hat{a}_2 \\ \hat{a}_1 & & \hat{a}_0 & \hat{a}_1 \\ \hat{a}_0 & & & \hat{a}_0 \end{bmatrix},$$

where

$$S_1 = S(f_1, f_2), \quad S_2 = S_2(f_1, f_2), \quad S_3 = S_3(f_1, f_2), \quad S_4 = S_4(f_1, f_2).$$

The next section explains three preprocessing operations applied to the Sylvester matrix and its subresultant matrices.

5.4 The preprocessing operations applied to the Sylvester matrix

Although the Sylvester matrix and its subresultant matrices $S_k(f_1(x), f_2(x))$ of two exact polynomials $f_1(x)$ and $f_2(x)$ perform well in the GCD computations, they yield incorrect results in the presence of error [84]. More precisely, the computation of the GCD of $f_1(x)$ and $f_2(x)$ performed in floating point arithmetic leads to two inexact polynomials $g_1(x)$ and $g_2(x)$ that are coprime because their Sylvester matrix has full rank, even if $f_1(x)$ and $f_2(x)$ are not coprime [88]. Therefore the Sylvester and subresultant matrices $S_k(g_1(x), g_2(x))$ of two polynomials whose coefficients are perturbed must be preprocessed before an AGCD computation [84].

This section reviews the preprocessing operations that have been described and implemented in [39, 84] using the Sylvester matrix in power basis. Similar operations have been implemented in [91] for the Sylvester matrix in Bernstein basis. Mainly, three preprocessing operations have been investigated by Winkler et al [84]. The first operation normalises the coefficients of the polynomials using the geometric mean. The second operation is the replacement of $g_2(x)$ by $\alpha g_2(x)$, where α is a parameter to be determined. A substitution is the last operation whose aim is to reduce the ratio of the maximum element, in magnitude, of $S(g_1, g_2)$ to the minimum element, in magnitude, of $S(g_1, g_2)$, [84]. These preprocessing operation are considered in Sections 5.4.1, 5.4.2 and 5.4.3. The importance of these operations is pointed out in [86].

Initially, let two exact univariate polynomials $f_1(x)$ and $f_2(x)$ as defined in Eq. 3.1 be perturbed by noise, such that $g_1(x)$ and $g_2(x)$ are two inexact univariate polynomials as defined in Definition 3.4. However, two inexact univariate polynomials can be represented as the following:

$$g_1(x) = \sum_{i=0}^{m} a_{m-i} x^{m-i}, \quad a_m \neq 0, \tag{5.11}$$

$$g_2(x) = \sum_{j=0}^{n} b_{n-j} x^{n-j}, \quad b_n \neq 0. \tag{5.12}$$

The Sylvester matrix $S_k(g_1(x), g_2(x))$ is formed by the polynomials in Eqs. 5.11 and 5.12, and will be preprocessed based in the recent development in the AGCD computation, [88]. The following sections introduce the three preprocessing operations required prior to the computation of AGCD.

5.4.1 Normalisation by geometric mean

Recall from Chapter 4 that the kth Sylvester matrix $S_k(g_1(x), g_2(x))$ is represented as a partitioned structure matrix of two polynomials $g_1(x)$ and $g_2(x)$, where the coefficients of $g_1(x)$ lie in the first $n-k+1$ columns and the coefficients of $g_2(x)$ lie in the last $m-k+1$ columns. This makes $S_k(g_1(x), g_2(x))$ unbalanced, if $g_1(x)$ and $g_2(x)$ are not normalised, especially if the coefficients of $g_1(x)$ are much larger or smaller than the coefficients of $g_2(x)$ [84]. Therefore a prior normalisation process of each polynomial $g_1(x)$ and $g_2(x)$ must be applied to overcome the computational problems.

Several studies attempt to use the 2-norm of the coefficients [9, 20] and the geometric mean of the coefficients [87] for normalisation of the coefficients of a polynomial. However, it can be shown that normalisation by geometric mean instead of 2-norm yields a more balanced $S_k(g_1(x), g_2(x))$, [39, 87]. Moreover, the geometric mean is also selected due to its accuracy against any tiny change in the polynomial coefficients, in contrast to $1-, 2-$ and ∞ norms which omit this change, [21]. Under these circumstances, the pair of inexact univariate polynomials will be redefined as:

$$g_1(x) = \sum_{i=0}^{m} \bar{a}_{m-i} x^{m-i}, \quad \bar{a}_{m-i} = \frac{a_{m-i}}{\prod_{i=0}^{m} |a_{m-i}|^{\frac{1}{m+1}}}, \tag{5.13}$$

$$g_2(x) = \sum_{j=0}^{n} \bar{b}_{n-j} x^{n-j}, \quad \bar{b}_{n-j} = \frac{b_{n-j}}{\prod_{j=0}^{n} |b_{n-j}|^{\frac{1}{n+1}}}, \tag{5.14}$$

where a_i and b_i are the non-normalised coefficients. It is important to note that the geometric mean here is considering only the non-zero coefficients $a_i \neq 0$ and $b_i \neq 0$ and not all the coefficients [84]. The next section introduces the second preprocessing operation.

5.4.2 Relative scaling of polynomials

This operation introduces an arbitrary non-zero scalar α which satisfies

$$\text{GCD}(g_1(x), g_2(x)) \sim \text{GCD}(g_1(x), \alpha g_2(x)), \quad \alpha \in \mathbb{R}\backslash 0, \tag{5.15}$$

where $\sim$ denotes equivalence, because the GCD of two polynomials is defined to within an arbitrary non-zero scalar [84]. In the case of the GCD of exact polynomials $f_1(x)$ and $f_2(x)$, the scalar weight α can be set equal to one, unlike the inexact case of polynomials $g_1(x)$ and $g_2(x)$ where α must be included to obtain good computational results [39]. In particular, different values of α result in different AGCDs. Based on the normalised Eqs. 5.13 and 5.14, a well-defined parameter α is the relative weight of $g_2(x)$ to the unit weight of $g_1(x)$, [84].
The parameter α was first introduced by [86] who indicated that a well-determined rank of Sylvester matrix $S_k(g_1(x), \alpha g_2(x))$ is difficult to obtain by using random values of α. Therefore the procedure of choosing a non-zero scalar α will decide how good the results are. Furthermore, an AGCD computation requires an optimal value of α, which will be discussed in Section 5.4.4.
The next section introduces the third preprocessing operation.

5.4.3 A change in the independent variable

A significant variation in the magnitude of the coefficients of the polynomial can lead to unreliable computations [30]. Therefore the third preprocessing operation scales the independent variable x, to minimise the ratio between the maximum coefficient of $g_1(x)$ and $g_2(x)$, in magnitude, to the minimum coefficient of the $g_1(x)$ and $g_2(x)$, in magnitude [84]. This scaling is achieved by a substitution in

Eq. 5.16:

$$x = w\theta, \tag{5.16}$$

where w is the new independent variable and θ is a coefficient change value is to be determined [84]. Introducing parameter θ in Eq. 5.16 to the GCD computation of $f_1(x)$ and $f_2(x)$ does not change its degree. That is:

$$\text{GCD}(f_1(x), f_2(x)) = \text{GCD}(f_1(w\theta), f_2(w\theta)).$$

Accordingly, after introducing the parameter θ, the polynomials $g_1(x)$ and $g_2(x)$ in Eqs. 5.13 and 5.14 are represented as:

$$\bar{g}_1(w, \theta) = \sum_{i=0}^{m} (\bar{a}_{m-i}\theta^{m-i})w^{m-i}, \tag{5.17}$$

$$\bar{g}_2(w, \theta) = \sum_{j=0}^{n} (\bar{b}_{n-j}\theta^{n-j})w^{n-j}. \tag{5.18}$$

The method to find the parameter θ of one polynomial has been studied in [30, 87], and extended into two polynomials in [39]. In order to compute an AGCD of $S_k(\bar{g}_1(w, \theta), \alpha\bar{g}_2(w, \theta))$, optimal values α_0 and θ_0 of α and θ will be considered in the next section.

5.4.4 Linear programming

A computation of the optimal values α_0 and θ_0 of α and θ, respectively, is required in order to compute an AGCD of $S_k(\bar{g}_1(w, \theta), \alpha\bar{g}_2(w, \theta))$ of two polynomials $\bar{g}_{1_\theta}(w)$ and $\bar{g}_{2_\theta}(w)$ whose entries are $\{\bar{a}_{m-i}\theta^{m-i}\}_{i=0}^{m}$ and $\{\bar{b}_{n-j}\theta^{n-i}\}_{j=0}^{n}$, respectively. The optimal values of α_0 and θ_0 can be determined using the minimisation problem developed in [88]:

$$\alpha_0, \theta_0 = \arg \min_{\alpha,\theta} \left\{ \frac{\max\{\max_{i=0,\ldots,m} |\bar{a}_{m-i}\theta^{m-i}| \, , \max_{j=0,\ldots,n} |\alpha\bar{b}_{n-j}\theta^{n-j}|\}}{\min\{\min_{i=0,\ldots,m} |\bar{a}_{m-i}\theta^{m-i}| \, , \min_{j=0,\ldots,n} |\alpha\bar{b}_{n-j}\theta^{n-i}|\}} \right\}. \tag{5.19}$$

In particular, the minimisation problem can be expressed as:

Minimise $\frac{\lambda}{\psi}$

Subject to

$$\begin{aligned}
\lambda &\geqslant |\bar{a}_{m-i}|\theta^{m-i}, \quad i = 0, \cdots, m \\
\lambda &\geqslant \alpha|\bar{b}_{n-j}| \; \theta^{n-j}, \quad j = 0, \cdots, n \\
\psi &\leq |\bar{a}_{m-i}|\theta^{m-i}, \quad i = 0, \cdots, m \\
\psi &\leq \alpha|\bar{b}_{n-j}| \; \theta^{n-j}, \quad j = 0, \cdots, n \\
\psi &> 0, \qquad \alpha > 0, \qquad \theta > 0
\end{aligned}$$

On the assumption of the following:

$$\Lambda = \log\lambda, \quad \Psi = \log\psi, \quad \Theta = log\theta, \quad \nu = \log\alpha, \quad \bar{A}_i = \log|\bar{a}_{m-i}|,$$

and

$$\bar{B}_j = \log|\bar{b}_{n-j}|,$$

the minimisation problem can be expressed as:

Minimise $\Lambda - \Psi$

Subject to

$$\begin{aligned}
\Lambda - (m-i)\Theta &\geqslant \bar{A}_i, \qquad i = 0, \cdots, m \\
\Lambda - (n-j)\Theta - \nu &\geqslant \bar{B}_j, \qquad j = 0, \cdots, n \\
-\Psi + (m-i)\Theta &\geqslant -\bar{A}_i, \qquad i = 0, \cdots, m \\
-\Psi + (n-j)\Theta + \nu &\geqslant -\bar{B}_j, \qquad j = 0, \cdots, n.
\end{aligned}$$

The problem is solved using linear programming (LP) [87], based on the following expression

$$\Lambda - \Psi = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \Lambda \\ \Psi \\ \nu \end{bmatrix} \tag{5.20}$$

Consequently, the polynomials shown in Eq. 5.17 and Eq. 5.18 can be written as:

$$\bar{g}_1(w,\theta_0) = \sum_{i=0}^{m} (\bar{a}_{m-i}\theta_0^{m-i})w^{m-i}, \quad \text{or} \quad \bar{g}_1(w,\theta_0) = \frac{\sum_{i=0}^{m}(a_{m-i}\theta_0^{m-i})w^{m-i}}{\prod_{i=0}^{m}|a_{m-i}\theta_0^{m-i}|^{\frac{1}{m+1}}}, \tag{5.21}$$

and

$$\bar{g}_2(w,\theta_0) = \sum_{j=0}^{n} (\bar{b}_{n-j}\theta_0^{n-j})w^{n-j}, \quad \text{or} \quad \bar{g}_2(w,\theta_0) = \frac{\sum_{i=0}^{n}(b_{n-j}\theta_0^{n-j})w^{n-j}}{\prod_{j=0}^{n}|b_{n-j}\theta_0^{j}|^{\frac{1}{n+1}}}. \tag{5.22}$$

where α_0 and θ_0 are the calculated optimal values [87].

The Sylvester matrix $S_k(g_1(x), g_2(x))$ of two inexact polynomials $g_1(x)$ and $g_2(x)$ that were defined in Eq. 5.11 and Eq. 5.12 is therefore transformed to preprocessed $S_k(\bar{g}_1(w,\theta), \alpha\bar{g}_2(w,\theta))$ of two inexact polynomials $\bar{g}_1(w,\theta)$ and $\bar{g}_2(w,\theta)$ as defined in Eqs. 5.21 and 5.22, respectively [88]. The new Sylvester matrix is given by:

$$S(\bar{g}_1(w,\theta), \alpha\bar{g}_2(w,\theta)) =$$

$$\left[\begin{array}{ccccc|ccccc}
\bar{a}_m{\theta_0}^m & & & & & \alpha_0\bar{b}_n{\theta_0}^n & & & & \\
\bar{a}_{m-1}{\theta_0}^{m-1} & \bar{a}_m{\theta_0}^m & & & & \alpha_0\bar{b}_{n-1}{\theta_0}^{n-1} & \alpha_0\bar{b}_m{\theta_0}^n & & & \\
\vdots & \bar{a}_{m-1}{\theta_0}^{m-1} & \ddots & & & \vdots & \alpha_0\bar{b}_{n-1}{\theta_0}^{n-1} & \ddots & & \\
\vdots & \vdots & \ddots & \bar{a}_m{\theta_0}^m & & \vdots & \vdots & \ddots & \alpha_0\bar{b}_n{\theta_0}^n & \\
\bar{a}_1\theta_0 & \vdots & \ddots & \bar{a}_{m-1}{\theta_0}^{m-1} & & \alpha_0\bar{b}_1\theta_0 & \vdots & \ddots & \alpha_0\bar{b}_{n-1}{\theta_0}^{n-1} & \\
\bar{a}_0 & \bar{a}_1\theta_0 & \ddots & \vdots & & \alpha_0\bar{b}_0 & \alpha_0\bar{b}_1\theta_0 & \ddots & \vdots & \\
 & \bar{a}_0 & \ddots & \bar{a}_1\theta_0 & & & \alpha_0\bar{b}_0 & \ddots & \alpha_0\bar{b}_1\theta_0 & \\
 & & & \bar{a}_0 & & & & & \alpha_0\bar{b}_0 &
\end{array}\right] \tag{5.23}$$

[39]. The computation of an AGCD is implemented based on the preprocessed Sylvester matrix and its subresultant matrices $S_k(\bar{g}_1(w,\theta), \alpha\bar{g}_2(w,\theta))$, where k represents the order of the subresultant matrices. The next example illustrates how the preprocessing operations minimise the ratio between the coefficients of Sylvester matrix of two inexact polynomials.

Example 5.2. Let $F \in \mathbb{R}^{5\times5}$ be a matrix represent an exact image $\mathcal{F}$ as below:

$$F = \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 & 0.1419 & 0.6557 \\ 0.9058 & 0.2785 & 0.9706 & 0.4218 & 0.0357 \\ 0.1270 & 0.5469 & 0.9572 & 0.9157 & 0.8491 \\ 0.9134 & 0.9575 & 0.4854 & 0.7922 & 0.9340 \\ 0.6324 & 0.9649 & 0.8003 & 0.9595 & 0.6787 \end{bmatrix},$$

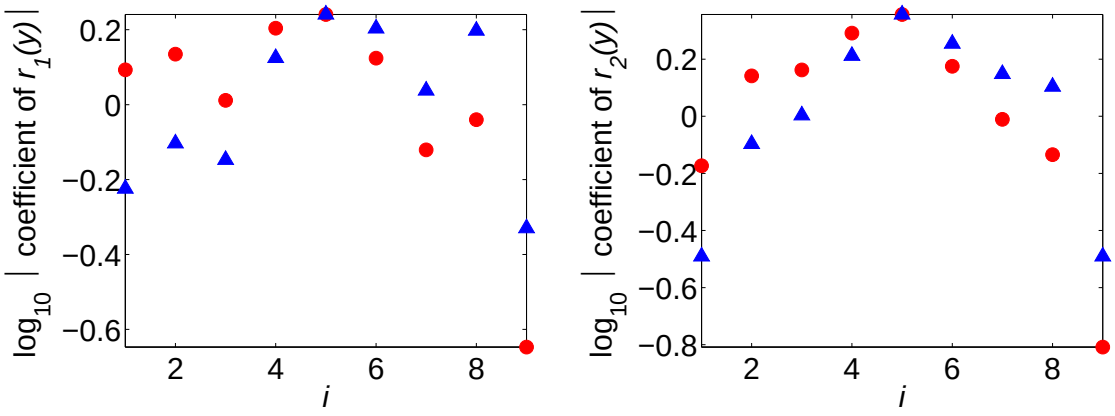
and $P \in \mathbb{R}^{3\times5}$ a separable PSF

$$P = \begin{bmatrix} 0.7060 \\ 0.0318 \\ 0.2769 \end{bmatrix} \begin{bmatrix} 0.7577 & 0.7431 & 0.3922 & 0.6555 & 0.1712 \end{bmatrix}.$$

Introducing the errors 10^{-4} and 10^{-6} as $\mathcal{E}$ and $\mathcal{N}$ respectively into the blurring model in Eq. 4.1, yield

$$G = \begin{bmatrix} 0.4359 & 0.4797 & 0.3611 & 0.5627 & 0.6125 & 0.4681 & 0.2663 & 0.3206 & 0.0793 \\ 0.5043 & 0.6459 & 0.9325 & 1.2567 & 0.7752 & 0.6395 & 0.3344 & 0.0820 & 0.0079 \\ 0.2608 & 0.5755 & 1.0172 & 1.4786 & 1.7423 & 1.4198 & 0.8937 & 0.6325 & 0.1339 \\ 0.6818 & 1.2526 & 1.4120 & 1.9036 & 1.9628 & 1.3469 & 0.8454 & 0.5772 & 0.1192 \\ 0.3870 & 1.0336 & 1.4825 & 2.0264 & 2.2594 & 1.6301 & 1.0635 & 0.6515 & 0.1274 \\ 0.2069 & 0.4272 & 0.4482 & 0.6033 & 0.7015 & 0.4618 & 0.3011 & 0.2265 & 0.0480 \\ 0.1327 & 0.3326 & 0.4352 & 0.5857 & 0.6320 & 0.4349 & 0.2858 & 0.1687 & 0.0322 \end{bmatrix},$$

where its bivariate polynomial $g(x,y)$ is of degrees 8 and 6 with respect to x and y, respectively. Matrix P can be reconstructed from G, by choosing any two rows



(a) $r_1(y)$ **(b)** $r_2(y)$

Figure 5.1: The magnitude of the coefficients of $r_1(y)$ and $r_2(y)$, before (●) and after (▲) the preprocessing operations, for Example 5.2

and any two columns of G. Let consider

$$\begin{aligned} r_1(y) &= (0.0793)y^8 + (0.3206)y^7 + (0.2663)y^6 + (0.4681)y^5 + (0.6125)y^4 \\ &+ (0.5627)y^3 + (0.3611)y^2 + (0.4797)y + (0.4359), \\ r_2(y) &= (0.0480)y^8 + (0.2265)y^7 + (0.3011)y^6 + (0.4618)y^5 + (0.7015)y^4 \\ &+ (0.6033)y^3 + (0.4482)y^2 + (0.4272)y + (0.2069). \end{aligned} \tag{5.24}$$

The Sylvester matrix $S(r_1(y), r_2(y))$ is of order 16×16, and the preprocessing operations of $S(r_1(y), r_2(y))$ can be seen in Figure 5.1, which show the coefficients of polynomials $r_1(y)$ and $r_2(y)$, before and after the preprocessing. This figure shows a drop-off in the variation magnitude of the coefficients of $r_1(y)$, as shown in (a), and the coefficients of $r_2(y)$, as shown in (b). The same is applied when columns are considered.

The implemented AGCD computation will be explained in Chapter 6, which is based on the GCD computation developed in [84].

5.5 Summary

This chapter has provided an overview of the the Sylvester matrix and its subresultant matrices and has explained their relation to the computation of the GCD. The computations in the presence of noise require a modification to the Sylvester matrix and its subresultant matrices. The work presented in [39, 84] suggested three preprocessing operations to be applied on the Sylvester matrix of two inexact polynomials, as described in this chapter. It follows that a computation of an AGCD can be carried out by the modified Sylvester resultant matrix. The next chapter will explain the tools used to compute the PSF.

Chapter 6

The AGCD computation

6.1 Introduction

As was outlined in Chapter 5, given a blurred image in noiseless conditions, the PSF can be computed by using the GCD computation of two exact polynomials that correspond to two rows or/and two columns of that blurred image. However, computing the GCD for the blurred image in the presence of noise is unreliable, and thus it is necessary to use an AGCD computation instead. In this case, two AGCD computations must be performed to restore the PSF of separable type. Furthermore, this method can be extended to restore random types of the PSF, including separable and nonseparable PSFs if two blurred versions containing similar PSFs of an image are given. Basically by performing the AGCD computations for each row or/and each column in the first blurred image and their corresponding in the second blurred image. The computation of an AGCD is based on the Sylvester matrix and its subresultant matrices whose inputs are two inexact univariate polynomials. The Sylvester resultant matrix and its subre-

sultant matrices are presented in Chapter 5 for univariate polynomials. Three preprocessing operations were performed on the Sylvester matrix to refine the solution.

This chapter explains the tools implemented to compute the PSF, based in the work proposed in [84]. The AGCD computation of two inexact univariate polynomials is based on the modified Sylvester resultant matrix, which is described in Chapter 5. The computations of the degree of the GCD and an AGCD will be described in Section 6.2 using two methods, the residual of an approximate linear algebra equation and first principal angle between subspaces [84]. The computations of the AGCD coefficients will be discussed in Section 6.3, using different methods.

6.2 The computation of the degree

This section considers the methods used to calculate the degree of an AGCD of two polynomials in the presence of noise. However, it is an essential to distinguish between the computation of the degree of the GCD and the degree of an AGCD. Therefore Sections 6.2.1 and 6.2.3 present the methods used for both computations.

6.2.1 The degree of the GCD

The GCD of two polynomials $f_1(x)$ and $f_2(x)$ whose coefficients are known exactly is defined in Chapter 3. It is pointed out in Theorems 5.3, 5.4 and 5.5 that the GCD of $f_1(x)$ and $f_2(x)$ exists and equals the largest value of $k = 1, \ldots, \hat{d}$, such

that Eq. 5.9 has a solution

$$A_k \mathbf{x}_k = \mathbf{c}_k, \quad k = 1, \ldots, \min(m, n), \tag{6.1}$$

where $\mathbf{c}_k$ is the first column of S_k, [84]. It follows from Eq. 5.9 that

$$\begin{aligned} \mathcal{L}_k &\subset \mathcal{H}_k, \quad k = 1, \ldots, \hat{d}, \\ \mathcal{L}_k &\not\subseteq \mathcal{H}_k, \quad k = \hat{d}+1, \ldots, \min(m, n), \end{aligned} \tag{6.2}$$

where $\mathcal{L}_k$ and $\mathcal{H}_k$ are the spaces spanned by the column vector of $\mathbf{c}_k$ and the column space of A_k, respectively, [39, 88]. The main aim of this section is to compute the degree $\hat{d}$ of the GCD of two exact polynomials $f_1(x)$ and $f_2(x)$; that requires the largest value of k for which the column vector $\mathbf{c}_k$ lies in the column space A_k for $k = 1, \ldots, \hat{d}$ [84]. The two methods investigated are first principal angle between subspaces and residual of an approximate linear algebra equation, based on the work done in [84].

First principal angle

The degree of the GCD of two polynomials $f_1(x)$ and $f_2(x)$ whose coefficients are known exactly can be computed using the method of first principal angle ψ_k. This is based on the computation of the smallest angle ϕ_k between two spanned spaces $\mathcal{L}_k$ and $\mathcal{H}_k$ of the column vector $\mathbf{c}_k$ and the column space A_k respectively [31, 84]. It follows that

$$\phi_k = \min \angle(\mathcal{L}_k, \mathcal{H}_k) = \begin{cases} \epsilon_k = 0, & \text{for} \quad k = 1, \ldots, \hat{d}, \\ \epsilon_k > 0, & \text{for} \quad k = \hat{d}+1, \ldots, \min(m, n). \end{cases} \tag{6.3}$$

This equation shows that the angle ϕ_k between $\mathcal{L}_k$ and $\mathcal{H}_k$ increases after which $k = \hat{d}$. The largest value of k for $k = 1, \ldots, \hat{d}$ which satisfies $\phi_k = 0$ is equal to the degree $\hat{d}$ of the GCD of two polynomials $f_1(x)$ and $f_2(x)$

whose coefficients are known exactly, using Eq. 5.9, [31, 88].

Residual

The residuals r_k of the linear algebraic equation 6.1 is

$$r_k = \|A_k \mathbf{x}_k - \mathbf{c}_k\| = \begin{cases} \epsilon_k = 0, & \text{for} \quad k = 1, \ldots, \hat{d}, \\ \epsilon_k > 0, & \text{for} \quad k = \hat{d}+1, \ldots, \min(m,n). \end{cases} \tag{6.4}$$

This equation shows that the growth of residual r_k starts after which $k = \hat{d}$. Hence, the largest value of k for $k = 1, \ldots, \hat{d}$ which satisfies $\epsilon_k = 0$ is equal to the degree of the GCD of two polynomials $f_1(x)$ and $f_2(x)$ whose coefficients are known exactly [39, 84].

It is important to note that these methods perform well in a symbolic computing environment but fail when finite precision arithmetic is used [88]. Therefore, a prior step is required before applying these methods on a pair of inexact polynomials $g_1(x)$ and $g_2(x)$ to compute the degree of an AGCD. The computation of an AGCD is based on the preprocessed Sylvester matrix defined in Chapter 5. The following section shows the criteria that determine an optimal column $\mathbf{c}_k$, and how these can be reflected in the result, before the computational methods for the degree of an AGCD are addressed.

6.2.2 Optimal column selection

In Section 6.2.1 Winkler et al. [84] argued that, in the case of exact polynomials, the first column $\mathbf{c}_k(f_1)$ of the subresultant matrices of the Sylvester matrix $S_k(f_1, f_2)$ can be moved to the right hand side. Thus $\mathbf{c}_k(f_1)$ certainly lies in the

column space of $A_k(f_1, f_2)$ if f_1 and f_2 have a common divisor $\hat{d}(x)$ of degree k [88]. In the presence of noise, however, it is difficult to define the column vector $\mathbf{c}_k$ since $S_k(\bar{g}_1, \alpha_0\bar{g}_2)$, which is defined in Eq. 5.23, has full rank for each $k = 1, \ldots, \min(m, n)$. Thus a column in $S_k(\bar{g}_1, \alpha_0\bar{g}_2)$ that lies in the space spanned by the remaining columns of $S_k(\bar{g}_1, \alpha_0\bar{g}_2)$ does not exist [84]. It follows that Eq. 5.8 transforms to

$$A_{k,i}\mathbf{x}_{k,i} \approx \mathbf{c}_{k,i}, \quad k = 1, \ldots, \min(m, n);\ i = 1, \ldots, (m + n - 2k + 2), \tag{6.5}$$

where $\mathbf{c}_k$ is the i^{th} column of the $S_k(\bar{g}_1(w, \theta_0), \alpha_0\bar{g}_2(w, \theta_0))$, and $A_{k,i}$ is the matrix formed by the other columns of $S_k(\bar{g}_1, \alpha_0\bar{g}_2)$ [88],

$$A_{k,i} = [\mathbf{c}_{k,1}, \mathbf{c}_{k,2}, \ldots, \mathbf{c}_{k,i-1}, \mathbf{c}_{k,i+1}, \ldots, \mathbf{c}_{k,m+n-2k+1}, \mathbf{c}_{k,m+n-2k+2}].$$

The value of index i will decide whether $\mathbf{c}_{k,i}$ lies in $\mathbf{c}_{k,i}(\bar{g}_1)$ or $\mathbf{c}_{k,i}(\alpha_0\bar{g}_2)$ and $A_{k,i} = A_{k,i}(\bar{g}_1(w, \theta_0), \alpha_0\bar{g}_2(w, \theta_0))$, [84]. The determination of indices k and i are recorded when the error in the approximation 6.5 is a minimum [88]. Under these circumstances, an optimal column $\mathbf{c}_{k,i}$ of $S_k(\bar{g}_1(w, \theta_0), \alpha_0\bar{g}_2(w, \theta_0))$ will be determined if the following indices are computed: [27, 84],

- The degree $k = d$ of an AGCD of $\bar{g}_1$ and $\bar{g}_2$.
- The index $i = l$ of the column of $S_d(\bar{g}_1(w, \theta_0), \alpha_0\bar{g}_2(w, \theta_0))$, such that the column $c_{d,i}$ in the approximation 6.5 is satisfied.

In general, calculating an optimal index l and an optimal degree d of an AGCD of $\bar{g}_1(w, \theta_0)$ and $\bar{g}_2(w, \theta_0)$ gives better results than choosing an optimal value of $i = 1$, where the first column will be always from one side of $S(\bar{g}_1(w, \theta_0), \alpha_0\bar{g}_2(w, \theta_0))$, [39, 84].

The next section describes two methods for the degree calculation of an AGCD

of two inexact polynomials.

6.2.3 The degree of an AGCD

This section concerns the computation of the degree of an AGCD in the presence of noise. The AGCD definition of a pair of polynomials $g_1(x)$ and $g_2(x)$ whose coefficients are unknown is presented in Chapter 3. More precisely, the inexact nature of $g_1(x)$ and $g_2(x)$ causes the and subresultant matrices to have full rank. Therefore, the computation of the degree of an AGCD requires preprocessing operations,[87, 88], as described in Chapter 5. Consequently, the preprocessing operations of the Sylvester matrix $S(g_1(x), g_2(x))$ result in a new Sylvester matrix $S(\bar{g}_1(w, \theta_0), \alpha_0\bar{g}_2(w, \theta_0))$, which is represented in Eq. 5.23, based on a pair of preprocessed inexact polynomials that are defined in Eq. 5.21 and Eq. 5.22.
This section therefore modifies the methods explained in section 6.2.1, which involve the determination of the first principal angle between subspaces and residual of an approximation of linear algebra equation, with respect to the best values of index l and degree d of an AGCD [84, 88].

First principal angle

The degree of an AGCD of two polynomials $\bar{g}_1(w, \theta_0)$ and $\bar{g}_2(w, \theta_0)$ whose coefficients are unknown exactly can be calculated by the first principal angle $\psi_{k,i}$. In particular, the $\psi_{k,i}$ can be determined by calculating the smallest angle $\phi_{k,i}$ between the spaces $\mathcal{L}_{k,i}$ and $\mathcal{H}_{k,i}$ of the column vector $\mathbf{c}_{k,i}$ and the column space $A_{k,i}$, respectively, [31, 84], such that the following

equation 6.6 is satisfied:

$$\begin{aligned} \psi_{k,i} &= \min \angle(\mathcal{L}_{k,i}, \mathcal{H}_{k,i}), \\ k &= 1, \ldots, \min(m,n), \qquad i = 1, \ldots, l, \ldots, (m+n-2k+2), \end{aligned} \tag{6.6}$$

and

$$\begin{aligned} \mathcal{L}_{k,i} &= \text{span}\{\mathbf{c}_{k,i}\}, \\ \mathcal{H}_{k,i} &= \text{span}\{\mathbf{c}_{k,1}, \mathbf{c}_{k,2}, \ldots, \mathbf{c}_{k,i-1}, \mathbf{c}_{k,i+1}, \ldots, \mathbf{c}_{k,m+n-2k+1}\}. \end{aligned} \tag{6.7}$$

It is obvious that $\dim \mathcal{L}_{k,i} = 1$ and $\dim \mathcal{H}_{k,i} = m+n-2k+1$. The minimum value of angle $\phi_{k,i}$ of $\psi_{k,i}$ for each value of k is computed as

$$\phi_k = \min \left\{ \psi_{k,i} : i = 1, \ldots, l, \ldots, (m+n-2k+2) \right\}, \tag{6.8}$$

[84, 88]. It follows that the optimal column index $i = l_{\phi,k}$ is recorded for which each of the $\min(m,n)$ minima occurs, [39, 84]. Thus

$$\mathbf{l}_{\phi,k} = \begin{bmatrix} l_{\phi,1} & l_{\phi,2} & \ldots & l_{\phi,\min(m,n)} \end{bmatrix}. \tag{6.9}$$

The degree $k = d_\phi$ of an AGCD of polynomials $\bar{g}_1(w, \theta_0)$ and $\bar{g}_2(w, \theta_0)$, is equal to the index k for which the ratio ϕ_{k+1}/ϕ_k is a maximum, [39, 84]:

$$d_\phi = \left\{ k : \frac{\phi_{k+1}}{\phi_k} \longrightarrow \max, k = 1, \ldots, \min(m,n) - 1 \right\}, \tag{6.10}$$

where the column index $l_{\phi,d}$ is computed from Eq. 6.9. Then the column $\mathbf{c}_{k,i}$ in Eq. 6.5 can be defined using the indices $k = d_\phi$ and $i = l_{\phi,d_\phi}$ for which the error in this approximation is a minimum. The determination of d_ϕ can be found when a maximum ratio between two consecutive values of angle ϕ_k occurs, as shown in Eq. 6.10, in particular, when it changes significantly from a small to a large value [84, 88].

Residual

The residual $\mathbf{r}_{k,i}$ is another technique that has been implemented in [84] to define the optimal values of indices i and k of an AGCD. Given $\mathbf{z}_{k,i}$ the least squares (LS) solution of equation 6.5, the residual $\mathbf{r}_{k,i} = \mathbf{r}_{k,i}(A_{k,i}, \mathbf{c}_{k,i})$ in this case is defined as

$$\begin{aligned} \mathbf{r}_{k,i} &= \mathbf{c}_{k,i} - A_{k,i}\mathbf{z}_{k,i}, \\ A^{\dagger}_{k,i} &= (A^{T}_{k,i}A_{k,i})^{-1}A^{T}_{k,i}, \\ \mathbf{z}_{k,i} &= A^{\dagger}_{k,i}\mathbf{c}_{k,i}, \end{aligned} \tag{6.11}$$

where

$$k = 1, \ldots, \min(m, n), \qquad i = 1, 2, \ldots, \ldots, (m + n - 2k + 2),$$

and † denotes pseudo inverse [84, 88]. The minimum value of $\|\mathbf{r}_{k,i}\|$ is calculated for each subresultant matrix of order k, so that

$$r_k = \min_i \|\mathbf{r}_{k,i}\|. \tag{6.12}$$

[88]. The column index $i = l_{r,k}$ for which each of the minimum values $\|\mathbf{r}_{k,i}\|$ is recorded, thereby yields the array l_r: [84],

$$\mathbf{l}_r = \begin{bmatrix} l_{r,1} & l_{r,2} & \ldots & l_{r,\min(m,n)} \end{bmatrix}. \tag{6.13}$$

The degree $k = d_r$ of an AGCD of polynomials $\bar{g}_1(w, \theta_0)$ and $\bar{g}_2(w, \theta_0)$ is equal to the index k for which the ratio r_{k+1}/r_k is a maximum, [39, 84]:

$$d_r = \left\{ k : \frac{r_{k+1}}{r_k} \longrightarrow \max, k = 1, \ldots, \min(m, n) - 1 \right\}, \tag{6.14}$$

where the column index $\mathbf{l}_r$ is computed from Eq. 6.13. Then the column $\mathbf{c}_{k,i}$ in Eq. 6.5 can be defined using the indices $k = d_r$ and $i = l_{d_r}$ for which the error in this approximation is a minimum [88]. The determination of d_r

is seen when a maximum ratio between two consecutive values of residual r_k occurs, as shown in Eq. 6.14,[84].

The maximum change in the first principal angle and the residual in Eq. 6.9 and Eq. 6.13 respectively, are used instead of their minimum values [84]. The use of the ratio can be seen in the Example 6.1.

Example 6.1. [84]. Let $min(m, n) = 6$ and let $\mathbf{r} \in \mathbb{R}^6$ be the vector of residual r_k for $k = 1, \ldots, 6$,

$$\begin{aligned} \mathbf{r}: &= \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{bmatrix}, \\ &= \begin{bmatrix} 10^{-10} & 4 \times 10^{-11} & 6 \times 10^{-8} & 2 \times 10^{-10} & 10^{-2} & 10^{-1} \end{bmatrix}. \end{aligned}$$

Obviously, the value of r_2 represents the minimum residual and the variation in residuals $r_1, \ldots, r_4$ are relatively small, and thus their approximate solutions of Eq. 6.5 are acceptable. On the other hand, it is seen that $r_5 \ll r_4$, which means the least squares solutions of Eq. 6.5 give an unacceptably large error for $k = 5$. The degree d_r of an AGCD is equal to 5 based on the residual method.

In general, it has been shown that the results obtained from both methods are similar for the degree of an AGCD, with only a small difference in the column index of i [27, 39, 84].

The next section briefly explain the decomposition tools applied to the modified Sylvester matrix type, to calculate its rank.

6.2.4 Rank determination using QR decomposition

This section considers the QR decomposition tool that is implemented to simplify the computation and to decompose the Sylvester resultant and subresultant ma-

trices. The degree of the GCD is equal to the rank of Sylvester resultant matrix type. However, the degree of an AGCD of two inexact polynomials transforms the problem to the rank determination, basically by performing the SVD or QR decomposition on the modified Sylvester resultant matrix.

Following Definition 2.2 of the SVD, the rank is equal to the number of σ_i such that $\sigma_i \neq 0$ [11, 31]. Using the SVD tool to determine the proximity of a full rank matrix to a nearby rank deficient matrix is computationally expensive [25]. The rank determination of a Sylvester matrix that constructed using two polynomials of high degree will require intensive computing. Therefore a simpler tool is introduced – QR decomposition. This refers to the factorisation of a matrix into two matrices in a special form and was proposed in [10, 11, 59]. Suppose a matrix $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, and let $Q \in \mathbb{R}^{m\times m}$ be an orthogonal matrix. Then, the minimisation problem $\min_x \|A\mathbf{x}-\mathbf{b}\|_2$ is equivalent to the orthogonal transformation of the linear least square problem

$$\min_x \|Q^T(A\mathbf{x}-\mathbf{b})\|_2, \tag{6.15}$$

where the notation $\|\cdot\|$ is referred to the vector norm, and Eq. 6.15 preserves the Euclidean length [11, 31]. According to the QR decomposition of $A \in \mathbb{R}^{m\times n}, m \geq n$, there is an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ such that

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{6.16}$$

where R is an upper triangular matrix with non-negative elements in the diagonal [11].

The QR decomposition of matrix $A \in \mathbb{R}^{m\times n}$ is not unique if $\text{rank}(A) < n$ [11]. Moreover, the QR decomposition can be implemented in a certain way such that

deleting or inserting a column or a row of a matrix is acceptable. More precisely, the deletion procedure of a column or a row applied to the modified Sylvester matrix using QR decomposition does not require to be implemented iteratively on its subresultant matrices, because the information is already preserved [11, 31, 84]. The estimation of the degree of an AGCD using the QR decomposition method is much faster than the SVD with larger matrix sizes [59]. The experimental results in this research considers the QR decomposition on the modified Sylvester matrix of two univariate inexact polynomials, as it has been implemented in [84]. However, the QR decomposition is computationally much more efficient than SVD.

The next example illustrates how an AGCD degree of two inexact polynomials is computed.

Example 6.2. Considering the matrices and polynomials provided in Example 5.2. If two rows of G are defined in Eq. 5.24, and two columns of G are defined as below

$$\begin{aligned} c_1(x) &= (0.5857)x^6 + (0.6033)x^5 + (2.0264)x^4 + (1.9036)x^3 + (1.4786)x^2 \\ &+ (1.2567)x + (0.5627), \\ c_2(x) &= (0.0322)x^6 + (0.0480)x^5 + (0.1274)x^4 + (0.1192)x^3 + (0.1339)x^2 \\ &+ (0.0079)x + (0.0793). \end{aligned}$$

Then, the degree computation of an AGCD is performed on the preprocessed Sylvester matrices $S(\bar{r}_1(w), \alpha\bar{r}_2(w))$ and $S(\bar{c}_1(w), \alpha\bar{c}_2(w))$, where they scaled by α and θ. The order of the PSF can be estimated using:

$$d_1 = \deg \mathrm{AGCD}\big(c_1(x), c_2(x)\big), \quad \text{and} \quad d_2 = \deg \mathrm{AGCD}\big(r_1(y), r_2(y)\big).$$

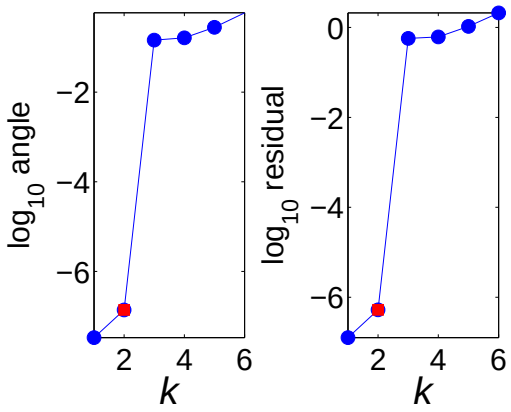


Figure 6.1: Row rank estimation, for Example 6.2

where d_1 and d_2 are the estimated degrees of the rows and columns of the PSF, respectively. Figures 6.1 and 6.2 demonstrate the rank estimation, using the first principal angle and the residual methods to calculate the degree of an AGCD of two inexact polynomials. It can be seen from Figure 6.1 that, the rank loss of $S(\bar{c}_1(w), \alpha\bar{c}_2(w))$ is equal to 2, and therefore, the PSF is of order 3 with respect to its rows, that $d_1 = 3$. While, the rank loss of $S(\bar{r}_1(w), \alpha\bar{r}_2(w))$ is equal to 4, as shown in Figure 6.2. More precisely, the degree of an AGCD of $r_1(y)$ and $r_2(y)$ is estimated as 5, and therefore, the PSF is of order 5 with respect to its columns, that $d_2 = 5$. These figures prove the equality of the obtained rank loss, using the residual and the first principal angle.

The computation of the coefficients of an AGCD is explained in the next section.

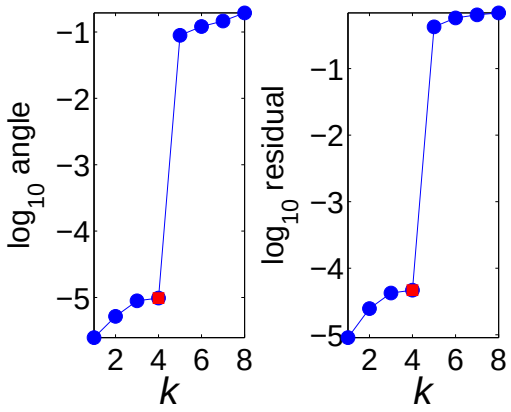


Figure 6.2: Column rank estimation, for Example 6.2

6.3 The computation of the coefficients of an AGCD

This section describes a method for the calculation of an AGCD polynomial of two inexact polynomials $g_1(x)$ and $g_2(x)$. The computation of an AGCD, here, is based on using the method of structured non-linear total least norms (SNTLN) applied to the modified Sylvester and subresultant matrices, [86, 87]. In particular, because the Sylvester matrix of inexact polynomials $g_1(x)$ and $g_2(x)$ has full rank, it is necessary to add a structured perturbation to the preprocessed Sylvester matrix in order to make it has rank deficient [87]. Then the coefficients of an AGCD can be calculated. Applying this method on Sylvester resultant matrix $S(\bar{g}_1(w,\theta_0), \alpha_0\bar{g}_2(w,\theta_0))$, which is defined in Eq. 5.23, will result in a structured low rank approximation $S(\tilde{g}_1(x), \tilde{g}_2(x))$, [86, 87]. However, the degree d of an AGCD of $g_1(x)$ and $g_2(x)$ has to be known first, as was illustrated in Section 6.2.3. It follows that the coefficients of an AGCD have to be calculated. The coefficients of an AGCD are calculated using three methods: least squares, QR,

and LU. The structured low rank approximation based on the SNTLN method and how it is related to an AGCD computation is addressed in the next section.

6.3.1 The computation of a structured low rank approximation

It has been shown in Theorem 5.2 that the degree of the GCD is equal to the rank loss of the Sylvester matrix of exact polynomials $f_1(x)$ and $f_2(x)$, and its coefficients are stated in the last non-zero row of the upper triangular factors of the reduced $S^T(f_1(x), f_2(x))$. It is assumed that $g_1(x)$ and $g_2(x)$, and therefore $\bar{g}_1(w)$ and $\bar{g}_2(w)$ are inexact and coprime, which causes their Sylvester matrix $S(\bar{g}_1(w), \bar{g}_2(w))$ to have full rank [87]. In this case, the computation of a structured low rank approximation requires small structured perturbations $z_{g_1}(w)$ and $z_{g_2}(w)$, which must be added to $\bar{g}_1(w)$ and $\bar{g}_2(w)$, respectively, to approximate the solution:

$$\begin{aligned} \tilde{g}_1(w) &= \sum_{i=0}^{m} (a_{m-i} + z_{g_1}) w^{m-i}, \quad i = 0, \ldots, m, \\ \tilde{g}_2(w) &= \sum_{j=0}^{n} (b_{n-j} + z_{g_2}) w^{n-j}, \quad j = 0, \ldots, n. \end{aligned}$$

and

$$\begin{aligned} \tilde{g}_1(w) &= \bar{g}_1(w) + z_{g_1}(w), \\ \tilde{g}_2(w) &= \bar{g}_2(w) + z_{g_2}(w), \end{aligned}$$

[87]. Specifically, the perturbed polynomials $\tilde{g}_1(w)$ and $\tilde{g}_2(w)$ tend to have a non-constant GCD [87], such that

$$\operatorname{rank} S(\tilde{g}_1(w), \tilde{g}_2(w)) = \operatorname{rank}\Big(S\big(\bar{g}_1(w), \bar{g}_2(w)\big) + S\big(z_{g_1}(w), z_{g_2}(w)\big)\Big) < (m+n). \tag{6.17}$$

Due to the structured nature of a Sylvester matrix, structured matrix methods can be used to compute the small structured perturbations $z_{g_1}(w)$ and $z_{g_2}(w)$. The calculation of $z_{g_1}(w)$ and $z_{g_2}(w)$ can be achieved using the SNTLN method [87].

The SNTLN method is described in Section 6.3.2, based on the method proposed in [87].

6.3.2 The method of SNTLN

The problem of calculating the coefficients of an AGCD of two inexact polynomials $g_1(x)$ and $g_2(x)$, can be formulated as the following:

Given the Sylvester matrix of $g_1(x)$ and $g_2(x)$, calculate the structured perturbation Sylvester matrix of $S(\bar{g}_1(w), \bar{g}_2(w))$ such that Eq. 6.17 is satisfied.

The polynomials $g_1(x)$ and $g_2(x)$ represent two rows or two columns form a degrade image/images. As aforementioned in Chapter 5 that preprocessing operations of the Sylvester resultant matrix $S(g_1(x), g_2(x))$ are required, which transfer the $S(g_1(x), g_2(x))$ to a new Sylvester resultant matrix $S(\bar{g}_1(w, \theta_0), \alpha_0 \bar{g}_2(w, \theta_0)$, as it defined in Eq. 6.21. It is important to notice that the constants α_0 and θ_0 are the initial values of α and θ [87]. It follows that a computation of the degree d of an AGCD of polynomials $\bar{g}_1(w, \theta_0)$ and $\bar{g}_2(w, \theta_0)$, defined in Eqs. 5.21 and 5.22 respectively, [87].

The d^{th} subresultant matrix $S_d(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0))$ of order

$$(m+n-d+1) \times (m+n-2d+2)$$

has full rank, because the $\bar{g_1}(w,\theta_0)$ and $\bar{g_2}(w,\theta_0)$ are inexact, as described in Section 6.2.3, [87]. Then, the homogeneous equation for $k=d$ and $i=l$,

$$S_d\big(\bar{g}_{1\theta}, \alpha\bar{g}_{2\theta}\big)\mathbf{w} = 0, \tag{6.18}$$

is reduced to the approximation equation $A\mathbf{x} \approx \mathbf{b}$ where $A \in \mathbb{R}^{(m+n-d+1)\times(m+n-2d+2)}$ after removing the optimal column l in $S_d(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0))$ to the right side [87]. This column is equal to $\mathbf{b}$ in Eq. 6.5, which is defined by the degree d. Considering $M_l \in \mathbb{R}^{(m+n-2d+1)(m+n-2d+1)}$ is the identity matrix after removing the column of index $i=l$, such that

$$M_l = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_{l-1} & \mathbf{e}_{l+1} & \dots & \mathbf{e}_{m+n-2d+1} & \mathbf{e}_{m+n-2d+2} \end{bmatrix}, \quad \mathbf{e}_i \in \mathbb{R}^{m+n-2d+1},$$

where $\mathbf{e}_i$ is the i^{th} unit basis vector and the l^{th} column of $S_d(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0))$ is equal to $S_d(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0))\mathbf{e}_l$.

Therefore, removing the column l of $S_d(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0))$ to the right hand side gives the approximate linear algebraic equation,

$$\Big(S_d\big(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0)\big)M_l\Big)\mathbf{x} \approx S_d\big(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0)\big)\mathbf{e}_l \quad \mathbf{x} \in \mathbb{R}^{m+n-2d+1}. \tag{6.19}$$

In order to define the perturbation that has to be added on the d^{th} subresultant of Sylvester matrix which minimises the approximation in Eq. 6.19. The Structured

perturbations matrix B_d of $S_d(\bar{g_1}(w,\theta_0), \alpha_0\bar{g_2}(w,\theta_0))$ is defined as follows

$$B_d(\alpha,\theta,z) = \left[\begin{array}{cccc|cccc} z_m\theta^m & & & & \alpha z_{m+n+1}\theta^n & & & \\ z_{m-1}\theta^{m-1} & z_m\theta^m & & & \alpha z_{m+n}\theta^{n-1} & \alpha z_{m+n+1}\theta^n & & \\ \vdots & z_{m-1}\theta^{m-1} & \ddots & & \vdots & \alpha z_{m+n}\theta^{n-1} & \ddots & \\ & \vdots & \ddots & z_m\theta^m & & \vdots & \ddots & \alpha z_{m+n+1}\theta^n \\ z_1\theta & & \ddots & z_{m-1}\theta^{m-1} & \alpha z_{m+2}\theta & & \ddots & \alpha z_{m+n}\theta^{n-1} \\ z_0 & z_1\theta & \ddots & \vdots & \alpha z_{m+1} & \alpha z_{m+2}\theta & \ddots & \vdots \\ & z_0 & \ddots & z_1\theta & & \alpha z_{m+1} & \ddots & \alpha z_{m+2}\theta \\ & & & z_0 & & & & \alpha z_{m+1} \end{array}\right],$$

where

$$z_{g_1}(\theta) = \left[\begin{array}{cccc} z_m\theta^m & z_{m-1}\theta^{m-1} & \ldots & z_1\theta \quad z_0 \end{array}\right]^T \in \mathbb{R}^{m+1},$$

$$\alpha z_{g_2}(\theta) = \left[\begin{array}{cccc} \alpha z_{m+n+1}\theta^n & \alpha z_{m+n}\theta^{n-1} & \ldots & \alpha z_{m+2}\theta \quad \alpha z_{m+1} \end{array}\right]^T \in \mathbb{R}^{n+1}.$$

It follows that $\mathbf{z}$ can be expressed as below

$$\mathbf{z} = \left[\begin{array}{cccccc} z_0 & \ldots & z_m & z_{m+1} & \ldots & z_{m+n+1} \end{array}\right]^T \in \mathbb{R}^{m+n+2}.$$

Similarly to S_d, the column $\mathbf{h}$ of index l in B_d is removed to the right side as is shown below

$$\Big(\Big(S_d(\alpha,\theta) + B_d(\alpha,\theta,z)\Big)M_l\Big)\mathbf{x} = \mathbf{c}_d(\alpha,\theta) + \mathbf{h}_d(\alpha,\theta,z), \tag{6.20}$$

where the quantities α, θ, $\mathbf{x}$ and $\mathbf{z}$ are to be calculated by the SNTLN method using Newton-Raphson method, and $\mathbf{c}_d$ and $\mathbf{h}_d$ are the l^{th} columns of $S_d(\alpha,\theta)$ and $B_d(\alpha,\theta,z)$ respectively [87], such that

$$\mathbf{c}_d(\alpha,\theta) = S_d(\alpha,\theta)\mathbf{e}_l \quad \text{and} \quad \mathbf{h}_d(\alpha,\theta,z) = B_d(\alpha,\theta,z)\mathbf{e}_l.$$

Winkler et al. [87] has pointed that depending on the column l the $\mathbf{c}_d$ and $\mathbf{h}_d$ may or may not be functions of α. It is assumed here that $\mathbf{c}_d$ and $\mathbf{h}_d$ are functions of

α if $n-d+2 \leq l \leq m+n-2d+2$, and not a functions of α if $1 \leq l \leq n-d+1$ [87]. Furthermore, Winkler et al. [87] has solved Eq. 6.20, and thus, the output after the preprocessing operations and derivations yields a structured low rank approximation of $S(g_1(x), g_2(x))$ of rank $m+n-d$. The SNTLN method is described in details in [87]. The corrected form of the polynomials $g_1(w)$ and $g_2(w)$ after applying the SNTLN method are $\tilde{g_1}(w)$ and $\tilde{g_2}(w)$, and thus, the Sylvester matrix of after preprocessing and the SNTLN methods is given by $S(\tilde{g_1}(w), \tilde{g_2}(w))$. This can be given as the following

$$S(\tilde{g}_1(w,\theta), \alpha\tilde{g}_2(w,\theta)) =$$

$$\left[\begin{array}{ccccc|ccccc}
\tilde{a}_m{\theta_0}^m & & & & & \alpha_0\tilde{b}_n{\theta_0}^n & & & & \\
\tilde{a}_{m-1}{\theta_0}^{m-1} & \tilde{a}_m{\theta_0}^m & & & & \alpha_0\tilde{b}_{n-1}{\theta_0}^{n-1} & \alpha_0\tilde{b}_m{\theta_0}^n & & & \\
\vdots & \tilde{a}_{m-1}{\theta_0}^{m-1} & \ddots & & & \vdots & \alpha_0\tilde{b}_{n-1}{\theta_0}^{n-1} & \ddots & & \\
\vdots & \vdots & \ddots & \tilde{a}_m{\theta_0}^m & & \vdots & \vdots & \ddots & \alpha_0\tilde{b}_n{\theta_0}^n & \\
\tilde{a}_1\theta_0 & \vdots & \ddots & \tilde{a}_{m-1}{\theta_0}^{m-1} & & \alpha_0\tilde{b}_1\theta_0 & \vdots & \ddots & \alpha_0\tilde{b}_{n-1}{\theta_0}^{n-1} & \\
\tilde{a}_0 & \tilde{a}_1\theta_0 & \ddots & \vdots & & \alpha_0\tilde{b}_0 & \alpha_0\tilde{b}_1\theta_0 & \ddots & \vdots & \\
 & \tilde{a}_0 & \ddots & \tilde{a}_1\theta_0 & & & \alpha_0\tilde{b}_0 & \ddots & \alpha_0\tilde{b}_1\theta_0 & \\
 & & & \tilde{a}_0 & & & & & \alpha_0\tilde{b}_0 &
\end{array}\right]. \tag{6.21}$$

where $\tilde{a_i}$ and $\tilde{b_j}$ are the corrected coefficients of $\tilde{g_1}$ and $\tilde{g_2}$ for $i = 0, \cdots, m$ and $j = 0, \cdots, n$ respectively.

The coefficients of the AGCD polynomial is calculated using the perturbed polynomials $\tilde{g_1}(w)$ and $\tilde{g_2}(w)$, that are defined as the following

$$\tilde{g}_1(w,\theta_0) = \sum_{i=0}^{m} (\tilde{a}_{m-i}\theta_0^{m-i})w^{m-i}, \quad \text{or} \quad \tilde{g}_1(w,\theta_0) = \sum_{i=0}^{m} \big((\bar{a}_{d,m-i} + z_{m-i})\theta_0^{m-i}\big)w^{m-i},$$

and

$$\tilde{g}_2(w,\theta_0) = \sum_{j=0}^{n} (\tilde{b}_{n-j}\theta_0^{n-j})w^{n-j}, \quad \text{or} \quad \tilde{g}_2(w,\theta_0) = \sum_{j=0}^{n} \big((\bar{b}_{d,n-j} + z_{m+n+1}\theta_0^{n-j}\big)w^{n-j}.$$

The next example provides an illustration of the implemented SNTLN method, [87].

Example 6.3. Considers the Sylvester matrices $S(r_1(y), r_2(y))$ and $S(c_1(x), c_2(x))$ each of order 16×16, such that the inexact polynomials $r_1(y)$, $r_2(y)$, $c_1(x)$, and $c_2(x)$ are defined in Examples 6.2 and 6.3. The rank determination process was performed on the modified Sylvester matrices $S(\bar{r}_1(w), \alpha\bar{r}_2(w))$ and $S(\bar{c}_1(w), \alpha\bar{c}_2(w))$ in Example 6.3, where the estimated PSF is of order 3×5.
In order to calculate the coefficients of the AGCD polynomials, the SNTLN method is applied first on the Sylvester matrices:

$$S(\bar{r}_1(w), \alpha\bar{r}_2(w)), \quad \text{and} \quad S(\bar{c}_1(w), \alpha\bar{c}_2(w)),$$

resulting in the corrected Sylvester matrices:

$$S(\tilde{r}_1(w), \alpha\tilde{r}_2(w)), \quad \text{and} \quad S(\tilde{c}_1(w), \alpha\tilde{c}_2(w)),$$

where $\tilde{r}_1(w)$, $\tilde{r}_2(w)$, $\tilde{c}_1(w)$, and $\tilde{c}_2(w)$ are the polynomials calculated using the SNTLN method. Figure 6.3 (a), shows the normalised singular values of the Sylvester matrices $S(r_1(y), r_2(y))$, $S(\bar{r}_1(w), \alpha\bar{r}_2(w))$, and $S(\tilde{r}_1(w), \alpha\tilde{r}_2(w))$. The same is shown in Figure 6.3 (b), for the Sylvester matrices that represent the columns. It is easily seen that, [87]:

$$\text{rank}S(r_1(y), r_2(y)) = \text{rank}S(\bar{r}_1(w), \alpha\bar{r}_2(w)) = \text{rank}S(\tilde{r}_1(w), \alpha\tilde{r}_2(w)) = 12,$$

and

$$\text{rank}S(c_1(x), c_2(x)) = \text{rank}S(\bar{c}_1(w), \alpha\bar{c}_2(w)) = \text{rank}S(\tilde{c}_1(w), \alpha\tilde{c}_2(w)) = 10.$$

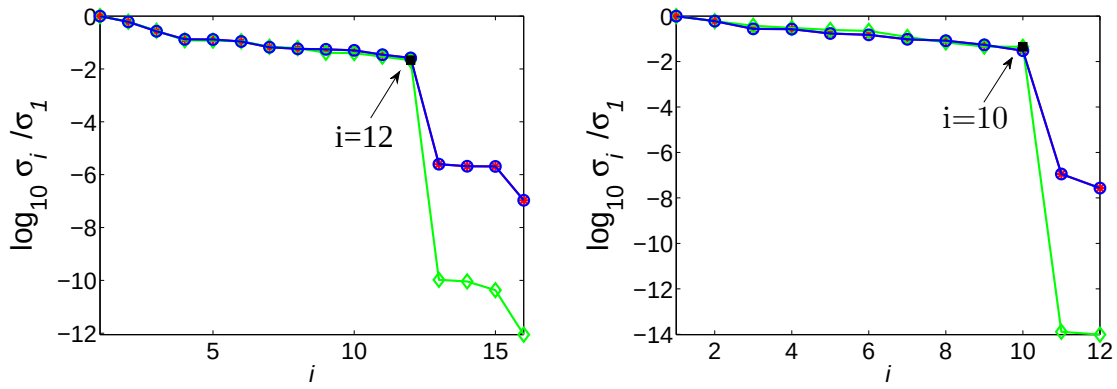


Figure 6.3: The normalised singular values of the Sylvester matrices: (a) $S(r_1(y), r_2(y))$ $\circ$; $S(\bar{r}_1(w), \alpha\bar{r}_2(w))$ $*$; $S(\tilde{r}_1(w), \alpha\tilde{r}_2(w))$ $\Diamond$, and (b) $S(c_1(x), c_2(x))$ $\circ$; $S(\bar{c}_1(w), \alpha\bar{c}_2(w))$ $*$; $S(\tilde{c}_1(w), \alpha\tilde{c}_2(w))$ $\Diamond$, for Example 6.3

The coefficients of the AGCD polynomial occupy the last nonzero row of the upper triangular factors R and U in QR and LU of the decomposed $S^T(\tilde{g}_1(w), \alpha\tilde{g}_2(w))$, [87]. Furthermore, the polynomial of the AGCD can be determined using least squares (LS) method.

The next section presents the LS method.

6.3.3 The least squares method

This section considers a method to calculate the coefficients of the AGCD polynomial using LS method. The LS method was also implemented previously in Bernstein basis to compute the AGCD polynomial after preprocessing operations [91]. The corrected polynomials $\tilde{g}_1(w, \theta)$ and $\tilde{g}_2(w, \theta)$ have a nonconstant common divisor of degree $\tilde{d}$, such that

$$\tilde{d}(w, \theta) = \frac{\tilde{g}_1(w, \theta)}{\tilde{u}_k(w, \theta)} = \frac{\alpha\tilde{g}_2(w, \theta)}{\tilde{v}_k(w, \theta)}, \tag{6.22}$$

and

$$\tilde{d} = \deg \mathrm{AGCD}(\tilde{g_1}(w,\theta), \tilde{g_2}(w,\theta)),$$

where $\tilde{u}_k(w,\theta)$ and $\tilde{v}_k(w,\theta)$ are the quotient polynomials.

The coefficients of $\tilde{u}_k(w,\theta)$ and $\tilde{v}_k(w,\theta)$ are obtained from the SNTLN method using the vector x, such that

$$\begin{bmatrix} \tilde{v}_0 \\ \vdots \\ \tilde{v}_{n-d}\theta_{n-d} \\ -\tilde{u}_0 \\ \vdots \\ -\tilde{u}_{m-d}\theta_{m-d} \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{q-1} \\ -1 \\ x_{q+1} \\ \vdots \\ x_{m+n-2d+2} \end{bmatrix} \in \mathbb{R}^{(m+n-2d+2)}.$$

It follows that

$$\tilde{u}(w,\theta_0) = \sum_{i=0}^{m-d} (\tilde{a}_{m-i}\theta_0^{m-i}) w^{m-i}, \qquad \text{and} \qquad \tilde{v}(w,\theta_0) = \sum_{j=0}^{n-d} (\tilde{b}_{n-j}\theta_0^{n-j}) w^{n-j}.$$

Using Eq. 6.22, the perturbed polynomials are given by

$$\tilde{g_{1_d}}(w,\theta_0) = \tilde{u}(w,\theta_0)\tilde{d}(w,\theta), \qquad \text{and} \qquad \alpha\tilde{g_{2_d}}(w,\theta_0) = \tilde{v}(w,\theta_0)\tilde{d}(w,\theta). \quad (6.23)$$

Alternately, Eqs. 6.23 can be written in the matrix form as the following

$$\begin{bmatrix} T_1[\tilde{u}(\theta)] \\ T_2[\tilde{v}(\theta)] \end{bmatrix} \tilde{d}(\theta) = \begin{bmatrix} \tilde{g_1}(\theta) \\ \alpha\tilde{g_2}(\theta) \end{bmatrix}, \quad (6.24)$$

where $T_1[\tilde{u}(\theta)] \in \mathbb{R}^{(m+1)(d+1)}$ and $T_2[\tilde{v}(\theta)] \in \mathbb{R}^{(n+1)(d+1)}$ are Toeplitz matrices,

$$T_1[\tilde{u}(\theta)] = \begin{bmatrix} \tilde{a}_{m-d} & & \\ \tilde{a}_{m-d-1} & \ddots & \\ \vdots & \ddots & \tilde{a}_{m-d} \\ \tilde{a}_0 & \vdots & \tilde{a}_{m-d-1} \\ & \ddots & \vdots \\ & & \tilde{a}_0 \end{bmatrix}, \quad T_2[\tilde{v}(\theta)] = \begin{bmatrix} \tilde{b}_{n-d} & & \\ \tilde{b}_{n-d-1} & \ddots & \\ \vdots & \ddots & \tilde{b}_{n-d} \\ \tilde{b}_0 & \vdots & \tilde{b}_{n-d-1} \\ & \ddots & \vdots \\ & & \tilde{b}_0 \end{bmatrix}.$$

and $\tilde{d}(\theta)$, $\tilde{g_1}(\theta)$, $\tilde{g_2}(\theta)$ are the vectors of scaled coefficients of $\tilde{d}(w,\theta)$, $\tilde{g_1}(w,\theta)$, $\tilde{g_2}(w,\theta)$. The coefficients of vector $\tilde{d}(\theta)$ can be calculated by solving the least squares problem of Eq. 6.24, such that

$$\tilde{d}(\theta) = \left(\begin{bmatrix} T_1[\tilde{u}(\theta)] \\ T_2[\tilde{v}(\theta)] \end{bmatrix} \right)^{\dagger} \begin{bmatrix} \tilde{g_1}(\theta) \\ \alpha\tilde{g_2}(\theta) \end{bmatrix}, \tag{6.25}$$

where † denoted to the Moore-Penrose pseudoinverse.

The coefficients of the AGCD polynomial $\tilde{d}(x)$ is obtained by transforming the coefficients of $\tilde{d}(\theta)$ from w variable into x variable,

$$\tilde{d_i}(x) = \frac{\tilde{d}(w,\theta)}{w\theta}, \quad i = 0, \cdots, d.$$

The LS, the QR and LU decomposition methods have been examined and compared. The obtained results shows that the LS method gives a small error compared with the other methods.

6.4 Summary

This chapter has described the two procedures applied on the modified Sylvester matrix and its subresultant matrices. First, the computation methods for the de-

gree of the GCD and an AGCD of two univariate polynomials are discussed and compared. The investigation found that both methods the first principal angle and the residual give a similar degree of an AGCD. Therefore one method will be implemented throughout the rest of this research, that is the residual method. The computation of the degree of an AGCD was performed based on two decomposition tools that are the SVD and the QR decomposition. However, the SVD method has some limitations including its failure to process large size of structured matrices, and its lack of an updating property, which exists in QR decomposition tool. Furthermore, the computation of an AGCD polynomial is considered at which the degree is defined. Followed by the computation of a structured low rank approximation. This computation requires to calculate the structured perturbation matrix $S(\delta\bar{g_1}(w), \delta\bar{g_2}(w))$ of Sylvester matrix $S(g_1(w), g_2(w))$ using the structured matrix methods SNTLN described on [87]. Three methods have been investigated to calculate the coefficients of an AGCD polynomial, including LS, QR and LU decompositions. The PSF estimation, therefore, can be obtained using the described method of AGCD that relies on the Sylvester resultant matrix. The PSF identification methods are described in the next chapter.

Chapter 7

The PSF identification using the AGCD computation

7.1 Introduction

The BID problem can be solved by employing polynomial computations. Chapter 4 has defined the BID in polynomial representation. This computation relies on the preprocessed Sylvester matrix, which is described in Chapter 5 and it is constructed from two univariate polynomials. In an image context, these polynomials represent two segments from a given degraded image or images, in particular, two rows or two columns. The PSF can be calculated using the GCD computation, as described in Chapter 6, such that the noise level is not given. This chapter discusses two situations in which the PSF is separable or nonseparable. Section 7.2 considers the estimation of a separable PSF. Section 7.3 considers the estimation of an arbitrary form of PSF, including separable and nonseparable.

7.2 The computation of separable PSF

Section 4.2 outlined the method of restoring a separable PSF. This method uses the AGCD computation described in Chapter 6, which is carried out by the modified Sylvester resultant matrix. Section 7.2.1 describes the calculation of a separable PSF using one degraded image. Section 7.2.2 describes the calculation of a separable PSF using two degraded images.

7.2.1 The PSF estimation using one degraded image

Consider an exact image $\mathcal{F}$ of order $M \times N$, and a PSF array $\mathcal{P}$ of order $C \times D$, where $C < M$ and $D < N$. It is assumed that a measurement error $\mathcal{E}$ is added to $\mathcal{P}$, such that the resulting PSF array is convolved with $\mathcal{F}$. Then noise $\mathcal{N}$ is added into the blurred image. The eventual degraded image $\mathcal{G}$ has the form of Eq. 4.1, such that

$$\mathcal{G} \in \mathbb{R}^{(M+C-1)\times(N+D-1)}. \tag{7.1}$$

The goal is to find $\mathcal{P}$ given the degraded image $\mathcal{G}$, without prior knowledge of the noise level.

As mentioned in Chapter 3, two methods were proposed to solve the BID problem using the Sylvester-type algorithm [24, 69]. One method uses the DFT in the GCD computation to minimise the computation time, however, this algorithm leads to error in the solution (e.g. quantization error in analog-to-digital conversion) [69]. While, the solution described in [24] is limited for a small blurring function of order 3×3 only, and the deblurring algorithm considers the first and the last rows, or/and columns, of the blurred image. Another drawback for this algorithm includes prior estimation of the noise level in the distorted image.

The work implemented in this chapter is able to determine the PSF that is perturbed by measurement error as shown in Eq. 7.1, using the described AGCD computation that can be performed without the use of DFT.

It was described above that if the convolved P is separable then its coefficients are formed by the multiplication of the vertical components, denoted by the vector $\mathbf{P}_c$, with the horizontal components, denoted by the vector $\mathbf{P}_r$, of the PSF, as shown in Eq. 2.12. In the case that $\mathbf{P}_r = 1$, then the PSF is represented as a 1D polynomial $p(x)$, such that $\mathbf{P} \in \mathbb{R}^{1\times D}$. Similarly, if $\mathbf{P}_c = 1$ then the PSF is represented as a 1D polynomial $p(y)$, such that $\mathbf{P} \in \mathbb{R}^{C\times 1}$. Algorithm 1 illustrates the method used to obtain a 1D PSF, by considering two rows using the described AGCD computation.

Algorithm 1 The 1D PSF estimation using one distorted image

Input
One distorted image $\mathcal{G}$.
Output
The 1D PSF, that is vector $\mathbf{P}$.
BEGIN

1. Read in a distorted image $\mathcal{G}$.
2. Choose two distinct rows $R = (r_1, r_2)$ of $\mathcal{G}$
3. Calculate their AGCD polynomial in row direction $p_r(y)$ which corresponds to vector $\mathbf{P}_r$:
$$p_r(y) = \text{AGCD}(r_1(y), r_2(y)). \tag{7.2}$$
4. Calculate the PSF, $\mathbf{P} = \mathbf{P}_r$.

END

The same applies for a 1D PSF array represented as a column vector $\mathbf{P}_c$, such that the AGCD computation considers two dissimilar columns of $\mathcal{G}$.

Example 7.1. Consider a true image $\mathcal{F}$ of size 70×100 pixels, and a 1D Gaussian PSF of dimensions 1×9 pixels, such that the PSF is given by the vector below:

$$\mathbf{P} = \begin{bmatrix} 0.0916, 0.1053, 0.1164, 0.1236, 0.1261, 0.1236, 0.1164, 0.1053, 0.0916 \end{bmatrix}.$$

A measurement error E of order 10^{-3} has been added to the vector $\mathbf{P}$, and noise N of order 10^{-5} has been added to the convolved resulting matrix. Figure 7.1 (a) shows the degraded image $\mathcal{G}$ of order 70×108 pixels. To calculate the 1D (i.e. row vector) PSF, two rows r_1 and r_2 of $\mathcal{G}$ are selected such that each contains the PSF information, and each row is represented as a univariate polynomial in y direction.

The degree of an AGCD of the two univariate polynomials is computed and is equal to 9, using the residual based method for the rank estimation, see Figure 7.1 (c). The normalized singular values of $S(\bar{r}_1(\theta, w), \alpha\bar{r}_2(\theta, w))$, $S(r_1(x), \alpha r_2(x))$, and $S(\tilde{r}_1(w), \alpha\tilde{r}_2(w))$ are illustrated in Figure 7.1 (e), where it is seen that its numerical rank is equal to 206, which indicates that the degree of the GCD of the corrected polynomials $\tilde{r}_1$ and $\tilde{r}_2$, from the SNTLN method, is 9.

The coefficients of the PSF are calculated using the LS method, where its inputs are $\tilde{r}_1$ and $\tilde{r}_2$. The exact and estimated 1D PSFs are shown in 7.1 (d), with an error in the GCD equal to $3.8789e-04$.

Since the PSF is known, it is possible to restore the exact image using polynomial division, as shown in Figure 7.1 (b). The method used to restore the exact image will be discussed in the next chapter.

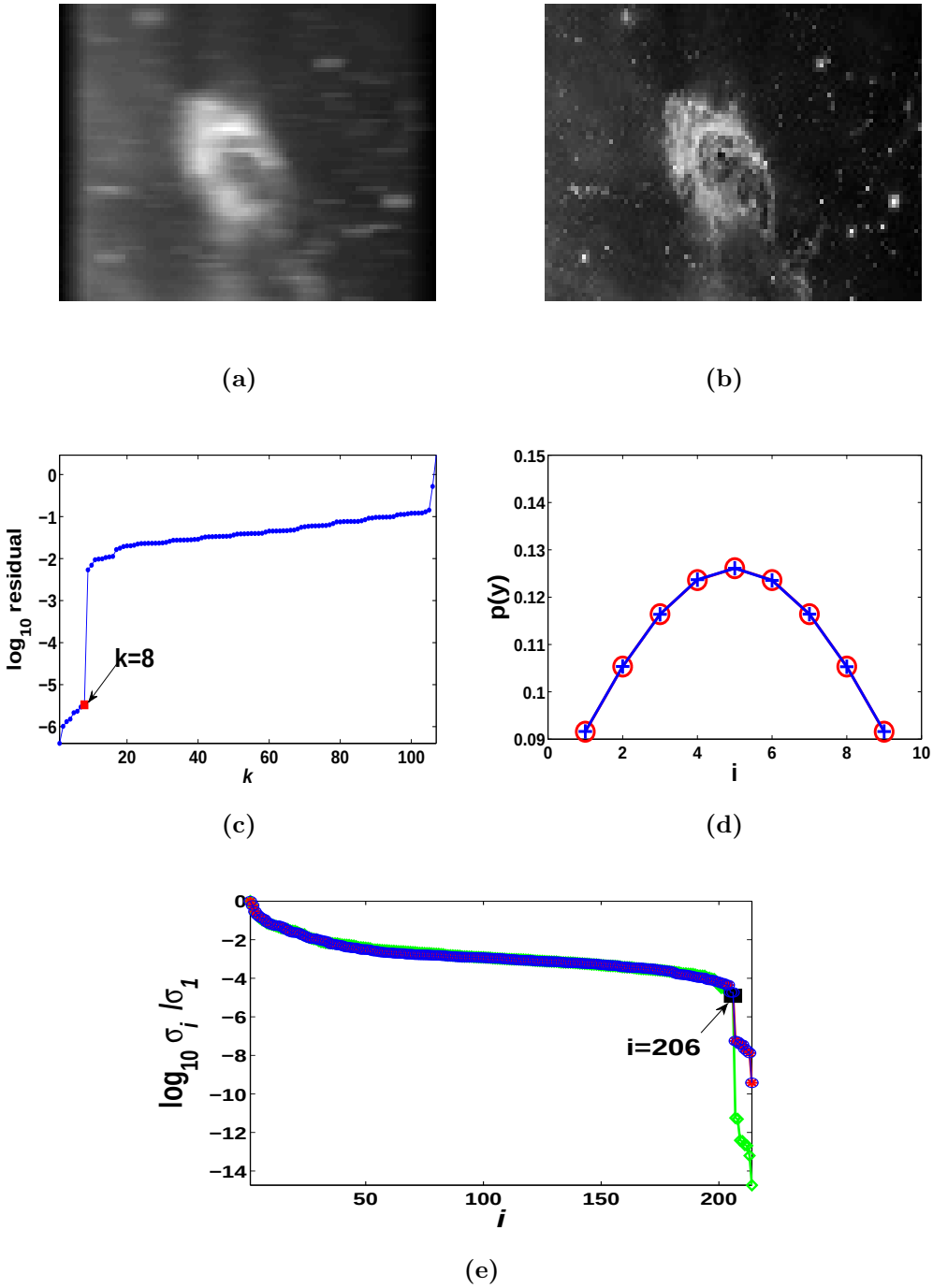


Figure 7.1: The results of Example 7.1, using the AGCD computation. (a) a degraded image; (b) a restored image ;(c) the estimated rank in row direction; (d) the exact PSF (o) and the estimated PSF ($+$); (e) the normalized singular values of $S(\bar{r}_1(\theta, w), \alpha\bar{r}_2(\theta, w))$ $*$, $S(r_1(x), \alpha r_2(x))$ $\circ$, and $S(\tilde{r}_1(w), \alpha\tilde{r}_2(w))$ $\Diamond$

Algorithm 2 describes the estimation method of a PSF giving one degraded image based on the AGCD computation algorithm [87].

Algorithm 2 The 1D PSF estimation using one degraded image

Input
One distorted image $\mathcal{G}$.
Output
The 2D PSF.
BEGIN

1. Read in a distorted image $\mathcal{G}$.
2. Choose two rows $R_1 = (r_1, r_2)$ and two columns $R_2 = (c_1, c_2)$ of $\mathcal{G}_2$.
3. Calculate their AGCDs $p_r(y)$ and $p_c(x)$, such that

$$p_r(y) = \text{AGCD}(r_1(y), r_2(y)), \quad \text{and} \quad p_c(x) = \text{AGCD}(c_1(x), c_2(x)). \tag{7.3}$$

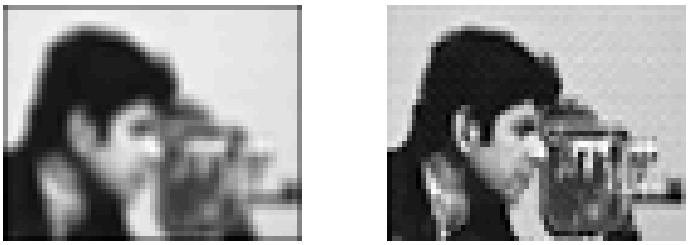
4. Calculate the PSF polynomial $p(x, y) = p_r(x)p_c(y)$, that corresponds to matrix P.

END

In addition, a 2D PSF is calculated using two AGCD computations. One considers two rows and the other considers two columns of the degraded image. Then the PSF is calculated from the two resulting AGCD polynomials in row and column directions, and their corresponding vectors are multiplied. This method was explained in Section 4.2, where one AGCD computation is performed on the first and last rows of $\mathcal{G}$ and another on the first and last columns of $\mathcal{G}$.

The next example discusses the results of the implemented BID algorithm compare to the method implemented in [24].

Example 7.2. Let consider a distorted image of dimensions 47×63 pixels, such that it is produced using 3×3 Gaussian PSF, measurement error $E = 10^{-5}$, and noise $N = 10^{-3}$. The work in [24] assumed the model in Eq 3.5, while the proposed model in this thesis is Eq 4.1. The second model is more challenging since the exact image is convolved with a corrupted PSF. However, applying the proposed algorithm on a distorted image, using the first model, can also solve the problem as shown in Figure 7.2, with GCD error of $4.423e-03$ in the blurring function.



(a) distorted image **(b)** Restored image

Figure 7.2: *The modified and restored images.*

Danelakis et al.'s method suggests that only one GCD operation is needed to compute the PSF, since the filter has the symmetry property. The top row is equal to the bottom row, while the left and right elements in the second row are copied from the middle element in the first row. The center of this PSF is calculated by subtracting the sum of total elements from 1.

The next section identifies the PSF from two degraded images.

7.2.2 The PSF estimation using two degraded images

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two dissimilar true images, each of dimension $M \times N$, and let $\mathcal{P}$ be the PSF array of dimension $C \times D$. Then $\mathcal{G}_1$ and $\mathcal{G}_2$ are two degraded versions of $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively, such that

$$\mathcal{G}_k = \mathcal{F}_k \otimes \big(\mathcal{P} + \mathcal{E}_k\big) + \mathcal{N}_k, \quad k = 1, 2$$

where $\mathcal{G}_k \in \mathbb{R}^{(M+C-1)\times(N+D-1)}$. Alternatively, the corresponding polynomial forms are expressed as

$$g_k(x, y) = f_k(x, y)\big(p(x, y) + e_k(x, y)\big) + n_k(x, y), \quad k = 1, 2.$$

If errors are not presented (i.e. $e_k(x, y) = 0$ and $n_k(x, y) = 0$ for $k = 1, 2$), then the GCD of g_1 and g_2 will lead to the exact PSF. However, if the errors are presented (i.e. $e_k(x, y) \neq 0$ and $n_k(x, y) \neq 0$ for $k = 1, 2$), then the AGCD computation of g_1 and g_2 leads to the PSF estimation. The 2D problem is simplified into two 1D cases, where one is considered in the row direction and the other in the column direction. The method is described in Chapter 4, where a row and a column are selected from each distorted image $\mathcal{G}_1$ and $\mathcal{G}_2$ such that each contains the whole information of the PSF.

$$h_k(x, y) \approx u_k(x, y)d(x, y) \quad k = 1, 2.$$

where $h_k(x, y)$ represents the degraded two rows or two column. An AGCD computation is performed using the row form $\mathcal{G}_1$ and its corresponding row from $\mathcal{G}_2$. Their AGCD is a univariate polynomial in row direction $d_r(y)$. The same is performed for the two columns, where their AGCD is a univariate polynomial in column direction $d_c(x)$. The two AGCDs polynomials are obtained by using the modified Sylvester resultant matrix, defined in Chapter 5, for each. Then the 2D

PSF is calculated from the resulting two AGCDs polynomials, using polynomial multiplication.

Algorithm 3 describes the method for the 2D separable PSF using two degraded images. The 2D separable PSF can be calculated using two consecutive AGCD

Algorithm 3 The PSF identification using two degraded image

Input
Two distorted images $\mathcal{G}_1$ and $\mathcal{G}_2$.
Output
The 2D PSF.
BEGIN

1. Read in two distorted images $\mathcal{G}_1$ and $\mathcal{G}_2$.
2. Choose a row r_1 and a column c_1 of $\mathcal{G}_1$ and their corresponding r_2 and a column c_2 of $\mathcal{G}_1$.
3. Calculate their AGCDs $d_r(y)$ and $d_c(x)$, such that

$$d_r(y) = \text{AGCD}(r_1(y), r_2(y)), \quad \text{and} \quad d_c(x) = \text{AGCD}(c_1(x), c_2(x)). \quad (7.4)$$

4. Calculate the PSF polynomial $d(x, y) = d_r(y)d_c(x)$, which corresponds to the approximated matrix $\tilde{P} = \tilde{\mathbf{P}}_r\tilde{\mathbf{P}}_c$, where $\tilde{\mathbf{P}}_r$ and $\tilde{\mathbf{P}}_c$ are the computed two vectors representing the horizontal and vertical components of the PSF.

END

computations. This section has provided a quick method to calculate the separable type of PSF; however, in many cases the separability condition of the PSF is not applied.

The procedures and methods to calculate an arbitrary form of PSF are discussed in the next section.

7.3 The computation of arbitrary PSF

It has been shown that the BID problem is an application of the GCD computations. A quick computation method was introduced above, to calculate an AGCD degree and its coefficients of a pair of inexact polynomials using a Sylvester resultant matrix. Also, it was shown that the PSF can be estimated using the described AGCD computation method, such that the noise and the PSF information are not required to be known a *priori*. This method is efficient in calculating separable types of the PSF.

It is then required to introduce a new method of BID, such that it includes an estimation of nonseparable types of PSF. Therefore the AGCD computation is extended to include iterative computations of AGCDs for each row and each column of two degraded images. This section aims to solve the following problem: Given a pair of inexact distorted images such that both are convolved with the same PSF, how can the AGCD computation be performed to restore arbitrary forms PSF, without prior knowledge of the PSF or the noise? A further discussion of the bivariate polynomials will be considered in Section 7.3.1. Section 7.3.2 considers a method of calculating the PSF using AGCD computations.

7.3.1 The GCD computation of bivariate polynomials

The GCD computation of bivariate polynomials are necessary for the solution of the BID problem, when the introduced PSF is not separable. Thus, this section explains the GCD computation when a pair of bivariate polynomials are given. First, it is important to distinguish between the GCD of two univariate polynomials and two bivariate polynomials. If two polynomials $f_1(x)$ and $f_2(x)$

of degrees m and n, respectively, are given such that $0 \leq n \leq m$, then there are two polynomials $q(x)$ and $r(x)$ that satisfy:

$$f_1(x) = f_2(x) \cdot q(x) + r(x), \quad r(x) \equiv 0 \quad \text{or} \quad \deg(r(x)) < n. \tag{7.5}$$

This equation has a different form in the case of bivariate polynomials. The implementation of Euclid's algorithm on univariate polynomials can be extended to bivariate polynomials. Let $f_1(x, y)$ and $f_2(x, y)$ be two bivariate polynomials. Then there are polynomials $p(y)$, $q(x, y)$, and $r(x, y)$ such that

$$p(y)f_1(x, y) = f_2(x, y) \cdot q(x, y) + r(x, y), \tag{7.6}$$

where $r(x, y) \equiv 0$ or $\deg r(x, y) < \deg f_2(x, y)$ with respect to x, and $p(y)$ is a polynomial of y only [12].

It follows that Eq. 7.6 can be written as a series of polynomial division as below:

$$\begin{aligned}
p_0(y)f_1(x, y) &= f_2(x, y) \cdot q_0(x, y) + r_1(x, y), \quad 0 < r_1(x, y), \\
p_1(y)f_2(x, y) &= r_1(x, y) \cdot q_1(x, y) + r_2(x, y), \quad 0 < r_2(x, y), \\
&\vdots \\
p_{j-1}(y)r_{j-2}(x, y) &= r_{j-1}(x, y) \cdot q_{j-1}(x, y) + r_j(x, y), \quad 0 < r_j(x, y), \\
p_j(y)r_{j-1}(x, y) &= r_j(x, y) \cdot q_j(x, y) + r_{j+1}(x, y),
\end{aligned} \tag{7.7}$$

where $\deg(r_1(x, y)) < \deg(f_2(x, y))$ with respect to x.

Consequently, the following are proved, using Eq. 7.7 in [12]:

- The polynomials $f_1(x, y)$ and $f_2(x, y)$ have a common divisor $r_{j+1}(x, y)$ if and only if $r_{j+1}(x, y) \equiv 0$, [12].

- The GCD of polynomials $f_1(x, y)$ and $f_2(x, y)$ is obtained if $r_{j+1}(x, y) \equiv 0$, by deleting all factors that are functions of y only from $r(x, y)$, [12].

It can be seen from the first equation in Eqs. 7.7 that $r_1(x, y)$ is a function of $f_1(x, y)$, $f_2(x, y)$, $q_0(x, y)$, and $p_0(y)$. The second equation in Eq. 7.7 shows that $r_2(x, y)$ can be represented as a function of $f_1(x, y)$, $f_2(x, y)$, $q_0(x, y)$, $q_1(x, y)$, $p_1(y)$, and $p_0(y)$ using the substitution of $r_1(x, y)$. Consequently, applying the same substitution procedure repeatedly on the remaining equations yields

$$r_{j+1}(x, y) = f_1(x, y)A(x, y) + f_2(x, y)B(x, y), \tag{7.8}$$

where $A(x, y)$ and $B(x, y)$ are polynomials in x and y.

As discussed above, there are two polynomials $\hat{u}(x, y)$ and $\hat{v}(x, y)$, such that

$$f_1(x, y) = \hat{u}(x, y)\hat{d}(x, y) \quad \text{and} \quad f_2(x, y) = \hat{v}(x, y)\hat{d}(x, y), \tag{7.9}$$

if and only if the polynomials $f_1(x, y)$ and $f_2(x, y)$ have a non-constant common divisor, that is $\hat{d}(x, y)$. The system of equations in Eq. 7.9 can be solved as below:

$$f_1(x, y)\hat{v}(x, y) - f_2(x, y)\hat{u}(x, y) = 0. \tag{7.10}$$

If $A(x, y) = \hat{v}(x, y)$ and $B(x, y) = -\hat{u}(x, y)$ and the polynomials $f_1(x, y)$ and $f_2(x, y)$ are not coprime, then $r_{j+1}(x, y) \equiv 0$.

The next section presents a method of calculating the PSF in the bivariate problem using an iterative GCD computation.

7.3.2 The iterative computation of GCD

This section considers the blur identification, $\mathcal{P}$, in Eq. 4.1. Chapter 4 defined the BID problem in the form of bivariate polynomials. The determination of $\mathcal{P}$ is an application of the GCD computation of two bivariate polynomials $g_1(x, y)$ and $g_2(x, y)$ of blurred images $\mathcal{G}_1$ and $\mathcal{G}_2$, as defined in Eq. 4.21, when $n_1(x, y) = n_2(x, y) = 0$ (i.e. noise-free) [69]. Let $g_1(x, y)$ and $g_2(x, y)$ be two bivariate

polynomials of blurred images $\mathcal{G}_1$ and $\mathcal{G}_2$ defined as below:

$$g_1(x,y) = \sum_{m=0}^{(M+C-2)} \sum_{n=0}^{(N+D-2)} g_1(m,n)x^m y^n, \tag{7.11}$$

$$g_2(x,y) = \sum_{m=0}^{(M+C-2)} \sum_{n=0}^{(N+D-2)} g_2(m,n)x^m y^n. \tag{7.12}$$

The bivariate polynomials $g_1(x,y)$ and $g_2(x,y)$ consist of $(M+C-1)$ univariate polynomials, in row direction.

$$g_1(m,y) = f_1(m,y)d(m,y), \quad m = 0,\cdots,M+C-2, \tag{7.13}$$

$$g_2(m,y) = f_2(m,y)d(m,y), \quad m = 0,\cdots,M+C-2. \tag{7.14}$$

It follows that

$$\mathrm{GCD}\big(g_1(m,y), g_2(m,y)\big) = d(m,y)\mathrm{GCD}\big(f_1(m,y), f_2(m,y)\big), \tag{7.15}$$

for $m = 0,\cdots,M+C-2$.

This equation defines a set of $(M+C-1)$ univariate GCD computations, resulting in $(M+C-1)$ polynomials $d(m,y)$ to be calculated along with the GCDs of $f_1(m,y)$ and $f_2(m,y)$. The degree of each $(M+C-1)$ GCD computation may vary, and thus prior initialisation for the length of each GCD vector is necessary. This can be achieved by appending zeros to the coefficients of each vector in the GCD computation, such that each vector has the length $N+D-1$.

Although the GCDs of the bivariate polynomials $f_1(x,y)$ and $f_2(x,y)$ are coprime, their reduction into univariate polynomials, when $x = \mu$, where μ is an arbitrary constant, may result in them having a non-constant GCD. For instance, consider two bivariate polynomials as below:

$$\begin{aligned} f_1(x,y) &= (x+2)(y+6x)(y-2x+1), \\ f_2(x,y) &= (x+1)(3y-2x)(y^2-x). \end{aligned}$$

It is clear that the polynomials $f_1(x, y)$ and $f_2(x, y)$ are coprime. However, their univariate polynomials when $x = 1$ are given as:

$$\begin{aligned} f_1(1, y) &= 3(y + 6)(y - 1), \\ f_2(1, y) &= 2(3y - 2)(y - 1)(y + 1), \end{aligned}$$

which have a non-constant GCD.

Therefore it is important to introduce Theorem 7.1, before proceeding with the GCDs of $f_1(m, y)$ and $f_2(m, y)$, as defined in Eq. 7.15. The following theorem is established in [85] using two bivariate polynomials $f_1(x, y)$ and $f_2(x, y)$, that are coprime.

Theorem 7.1. *If two polynomials $f_1(x, y)$ and $f_2(x, y)$ are coprime, and if the variable x equals an arbitrary constant $\mu \in \mathbb{C}$ located on the unit circle in $\mathbb{C}$, then the function $C(\mu, y)$ given by*

$$C(\mu, y) := GCD\big(f_1(\mu, y), f_2(\mu, y)\big),$$

with probability almost one, is independent of y [85].

Proof. Suppose that there are polynomials $A_1(x, y)$, $A_2(x, y)$, and $\psi_1(x)$; then from Eq. 7.8 the following equation is satisfied:

$$\psi_1(x) = f_1(x, y)A_1(x, y) + f_2(x, y)A_2(x, y), \tag{7.16}$$

where $\psi_1(x) \not\equiv 0$. Similarly, if there exist polynomials $B_1(x, y)$, $B_2(x, y)$, and $\psi_2(y)$, then

$$\psi_2(y) = f_1(x, y)B_1(x, y) + f_2(x, y)B_2(x, y), \tag{7.17}$$

where $\psi_2(y) \not\equiv 0$.

If the substitution of $x = \mu$ is applied to Eq. 7.17 then

$$\psi_2(y) = f_1(\mu, y)B_1(\mu, y) + f_2(\mu, y)B_2(\mu, y),$$

where μ is an arbitrary constant. Consider $C(\mu, y)$, the GCD of polynomials $f_1(\mu, y)$ and $f_2(\mu, y)$ such that

$$C(\mu, y) = \mathrm{GCD}\big(f_1(\mu, y), f_2(\mu, y)\big). \tag{7.18}$$

Subsequently, this leads the function $C(\mu, y)$ to be a factor of $\psi_2(y)$ for each μ and for which $\deg C(\mu, y) > 0$.Moreover, Eq. 7.16 satisfies

$$\psi_1(\mu) = f_1(\mu, y)A_1(\mu, y) + f_2(\mu, y)A_2(\mu, y), \tag{7.19}$$

such that $\psi_1(\mu) = 0$. More precisely, the right hand side of Eq. 7.19 is equal to zero if each value of y is a root $y = y_i(\mu)$ of $C(\mu, y)$, and thus $\psi_1(\mu) = 0$ for each value of μ that is connected to each value $y_i(\mu)$ of y.Consequently, every value of μ that satisfies $\deg C(\mu, y) > 0$ is a root of $\psi_1(x)$ as well.This means that the polynomial $C(\mu, y)$ is non-constant for limited number of values of x, since the polynomial $\psi_1(x)$ has a limited number of roots. The values of x are arbitrarily distributed in the complex plan $\mathbb{C}$. So it is assumed that these values of x are not located on the unite circle in $\mathbb{C}$, $|x| = 1$.The polynomial $C(\mu, y)$, however, is a constant for all the values of x that satisfy $|x| = 1$. This applies, in particular, when the value of μ is restricted to lie on the unit circle in $\mathbb{C}$. Therefore $C(\mu, y)$ is polynomial, and thus the polynomials $f_1(\mu, y)$ and $f_2(\mu, y)$ are coprime. □

The above theorem considers bivariate polynomials in the complex plane $\mathbb{C}$. Similarly, and without loss of generality the method explained in this work can be implemented on the real part $\mathbb{R}$. Thus the constant μ is equivalent to m in Eq. 7.15. Theorem 7.1 proves that the polynomials $f_1(m, y)$ and $f_2(m, y)$ in Eq.

7.15, with high probability, are coprime for each value of m, and therefore the computations of their GCDs are equal to the constant function $c(m)$, such that

$$\mathrm{GCD}\big(f_1(m,y), f_2(m,y)\big) = c(m), \quad m = 0, 1, \cdots, (M + C - 2). \tag{7.20}$$

It follows that the $M+C-1$ univariate polynomials of $d(m,y)$ can be calculated by using the substitution of Eq. 7.20 into Eq. 7.15:

$$\mathrm{GCD}\big(g_1(m,y), g_2(m,y)\big) = c(m)d(m,y), \quad m = 0, 1, \cdots, (M + C - 2). \tag{7.21}$$

It can be seen that the left hand side of Eq. 7.21 is proportional to the right hand side up to arbitrary scalars $c(m)$, for each value of m. The iterative computations of $(M + C - 1)$ GCD produce the scaled quantities $c(m)d(m,y)$, for $m = 0, 1, \cdots, (M + C - 2)$, such that the variable y is independent in each polynomial. It follows that the variable y in Eq. 7.20 is replaced by the values of n, such that

$$\mathrm{GCD}\big(g_1(m,n), g_2(m,n)\big) = c(m)d(m,n), \tag{7.22}$$

for $m = 0, 1, \cdots, (M + C - 2)$ and $n = 0, 1, \cdots, (N + D - 2)$. Then suppose the coefficients of the computed GCDs of two univariate polynomials $g_1(m,n)$ and $g_2(m,n)$ are stored in a matrix $A(m,n)$ of dimension $(M+C-1) \times (N+D-1)$. In particular, if

$$A(m,n) = \mathrm{GCD}\big(g_1(m,n), g_2(m,n)\big), \tag{7.23}$$

for $m = 0, 1, \cdots, (M+C-2)$ and $n = 0, 1, \cdots, (N+D-2)$, then the substitution into Eq. 7.23 yields the equation below:

$$\begin{aligned} A(m,n) = c(m)d(m,n), \quad m &= 0, 1, \cdots, (M + C - 2), \\ n &= 0, 1, \cdots, (N + D - 2). \end{aligned}$$

This equation can be represented as:

$$A(m,n)a(m) = d(m,n), \quad \begin{aligned} m &= 0,1,\cdots,(M+C-2), \\ n &= 0,1,\cdots,(N+D-2), \end{aligned} \tag{7.24}$$

where $a(m) = \frac{1}{c(m)}$.

This equation is obtained from the GCDs computations, that include rows of the distorted images. A similar equation is obtained when the computations of the GCDs for each column of the distorted images are considered, given by:

$$B(m,n)b(m) = d(m,n), \quad \begin{aligned} m &= 0,1,\cdots,(M+C-2), \\ n &= 0,1,\cdots,(N+D-2). \end{aligned} \tag{7.25}$$

Consequently, the relation between Eqs. 7.24 and 7.25 is provided by:

$$A(m,n)a(m) - B(m,n)b(m) = 0, \quad \begin{aligned} m &= 0,1,\cdots,(M+C-2), \\ n &= 0,1,\cdots,(N+D-2). \end{aligned} \tag{7.26}$$

In the matrix form, the $(M+C-1)(N+D-1)$ equations are expressed as

$$\begin{bmatrix} S_1 & -S_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 0, \tag{7.27}$$

where $S_1 \in \mathbb{R}^{[(M+C-1)(N+D-1)\times(M+C-1)]}$ and $S_2 \in \mathbb{R}^{[(M+C-1)(N+D-1)\times(N+D-1)]}$ are given by

$$S_1 = \begin{bmatrix} A(0,0) & 0 & 0 & \cdots & 0 \\ A(0,1) & 0 & 0 & \cdots & 0 \\ A(0,2) & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A(0,N+D-2) & 0 & 0 & \vdots & 0 \\ 0 & A(1,0) & 0 & \cdots & 0 \\ 0 & A(1,1) & 0 & \cdots & 0 \\ 0 & A(1,2) & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & A(1,N+D-2) & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & A(M+C-2,0) \\ 0 & 0 & 0 & \vdots & A(M+C-2,1) \\ 0 & 0 & 0 & \vdots & A(M+C-2,2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & A(M+C-2,N+D-2) \end{bmatrix},$$

$$S_2 = \begin{bmatrix} B(0,0) & 0 & 0 & \cdots & 0 \\ 0 & B(0,1) & 0 & \cdots & 0 \\ 0 & 0 & B(0,2) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & B(0,N+D-2) \\ B(1,0) & 0 & 0 & \cdots & 0 \\ 0 & B(1,1) & 0 & \cdots & 0 \\ 0 & 0 & B(1,2) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & B(1,N+D-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B(M+C-2,0) & 0 & 0 & \vdots & 0 \\ 0 & B(M+C-2,1) & \vdots & \vdots & \vdots \\ 0 & 0 & B(M+C-2,1) & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & B(M+C-2,N+D-2) \end{bmatrix},$$

and

$$\begin{aligned} y_1 &= \begin{bmatrix} a(0) & a(1) & \cdots & a(M+C-2) \end{bmatrix}^T \in \mathbb{R}^{M+C-1}, \\ y_2 &= \begin{bmatrix} b(0) & b(1) & \cdots & b(N+D-2) \end{bmatrix}^T \in \mathbb{R}^{N+D-1}. \end{aligned}$$

The solution of

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{7.28}$$

in Eq. 7.27 can be calculated by using the SVD computation of the coefficients matrix $S = \begin{bmatrix} S_1 & -S_2 \end{bmatrix}$. In particular, the solution of vector y is equal to the last column in the matrix V, where $S = U\Sigma V^T$ in $\mathbb{R}$.

Then the polynomial $d(m,n)$ is approximated using Eqs. 7.24 and 7.25 into

$$\begin{aligned} d(m,n) = \frac{1}{2}\big(A(m,n)a(m) + B(m,n)b(m)\big), \quad m &= 0,1,\cdots,(M+C-2), \\ n &= 0,1,\cdots,(N+D-2). \end{aligned} \tag{7.29}$$

This means that the approximated bivariate polynomial $d(m,n)$ is calculated by averaging the computed GCDs of the distorted images, once by considering their rows and again by considering their columns. The multiplication in Eqs. 7.24 and 7.25, and in particular the product of $A(m,n)a(m)$ and $B(m,n)b(m)$, can be described as:

$$\begin{aligned} A(0,n)a(0) &= d(0,n), \quad n = 0,1,\cdots,(N+D-2), \\ A(1,n)a(1) &= d(1,n), \quad n = 0,1,\cdots,(N+D-2), \\ A(2,n)a(2) &= d(2,n), \quad n = 0,1,\cdots,(N+D-2), \\ &\vdots \\ A(M+C-1,n)a(M+C-1) &= d(M+C-1,n), \quad n = 0,1,\cdots,(N+D-2), \end{aligned}$$

and

$$\begin{aligned}
B(m,0)b(0) &= d(m,0), \quad m = 0,1,\cdots,(M+C-2),\\
B(m,1)b(1) &= d(m,1), \quad m = 0,1,\cdots,(M+C-2),\\
B(m,2)b(2) &= d(m,2), \quad m = 0,1,\cdots,(M+C-2),\\
&\vdots\\
B(m,N+D-2)b(N+D-2) &= d(m,N+D-2), \quad m = 0,1,\cdots,(M+C-2),
\end{aligned}$$

respectively. The estimated PSF is then derived from Eq. 7.29, where its dimensions are already calculated in the first stage of the computation of the GCD.

This section has considered the iterative computation of the GCD, which estimates the PSF in two distorted images. The problem reduces to AGCD computation in the presence of noise. The AGCD computation method was described in Chapter 6.

Algorithm 4, shows the images deblurring method based on AGCD computations to calculate the PSF, either separable or nonseparable, from two distorted images.

Algorithm 4 Deblurring by AGCD computations and the HPC

Input Two distorted images $\mathcal{G}_1$ and $\mathcal{G}_2$.

Output The PSF, restored images $\tilde{\mathcal{F}}_1$ and $\tilde{\mathcal{F}}_2$.

Begin

1. Perform the univariate AGCD computations for each row of $\mathcal{G}_1$ and $\mathcal{G}_2$.

   - Initialisation of matrix A of dimensions $(M + C - 1) \times (N + D - 1)$, and vector k_1 of length $(M + C - 1)$.

   For $i = 1 : M + C - 1$

   (a) Define the ith rows of $\mathcal{G}_1$ and $\mathcal{G}_2$.
   $$r_1 = \mathcal{G}_1(i, :) \quad \text{and} \quad r_2 = \mathcal{G}_2(i, :)$$

   (b) Calculate the AGCD $d(i)$ of the two polynomials, that correspond to these row vectors.
   $$d(i) = \text{AGCD}(r_1, r_2)$$

   (c) Store the computed degree d_1 in ith entry of vector k_1.
   (d) Store the coefficients $d(i)$ in the ith row of matrix A.
   $$A(i, :) = d(i)$$

   End For

2. Repeat Step 5, by considering each columns of image $\mathcal{G}_1$ and each columns of image $\mathcal{G}_2$.

   - Initialisation of matrix B of dimensions $(M + C - 1) \times (N + D - 1)$, and vector k_2 of length $(N + D - 1)$.

   For $j = 1 : N + D - 1$

   (a) Define the jth columns of $\mathcal{G}_1$ and $\mathcal{G}_2$.
   $$c_1 = \mathcal{G}_1(:, j) \quad \text{and} \quad c_2 = \mathcal{G}_2(:, j)$$

   (b) Calculate the AGCD $d(j)$ of the two polynomials, that correspond to these column vectors.
   $$d(j) = \text{AGCD}(c_1, c_2)$$

   (c) Store the computed degree d_2 in jth entry of vector k_2.
   (d) Store $d(j)$ in the jth column of matrix B.
   $$B(:, j) = d(j)$$

   End For

3. Solve Eq. 7.27 for y_1 and y_2 using the calculation of the SVD of coefficients matrix S, that formed by S_1 and S_2.

4. Perform the calculation in Eq. 7.29, to obtain $d(m, n)$.

5. Find the most frequent degrees d_1 and d_2 in vectors k_1 and k_2 respectively.

6. Derive $\mathcal{P}$, that represent the 2D PSF, from $d(m, n)$ using the estimated dimensions d_1 and d_2.

7. Perform polynomial division to restore $\mathcal{F}_1$ and $\mathcal{F}_1$.

$$\tilde{f}_1(x, y) = g_1(x, y)/p(x, y) \quad \text{and} \quad \tilde{f}_2(x, y) = g_2(x, y)/p(x, y)$$

END

It can be seen that the output of each AGCD computation of univariate polynomials will include the estimation of the degree and its coefficients. These coefficients are used then to form the estimated PSF. It is important to note that some of the computed degree may vary, depending on the imposed noise and the coefficients. It follows that estimation of its coefficients is poor. Therefore an improvement to the previous method has been done, splitting the iterative computation of AGCD into two stages.

The first stage is to compute the degree of each univariate polynomial, by considering each row and each column of two distorted images. This works by finding the most frequently occurring degree(i.e. majority voting); in particular, this procedure is applied to the vectors that store the computed degrees for each row, and the vector that stores the degrees for each columns. So the computed degrees are d_1 and d_2, which form the dimensions of the computed PSF in the x and y directions respectively.

The second stage uses the estimated degree d_1 and d_2 to calculate the stored coefficients in each row and each column of two distorted images. The advantage of using this technique is that it gives more accurate results for the computed degree and the coefficients of AGCD.

In general, the difference between the two methods are as below.

Method 1: The computation of AGCD are applied to each univariate polynomial of two distorted images, such that the output from each iteration consists of the computed rank and its coefficients. In particular, the degree and its coefficients are calculated simultaneously in each iteration. The computed coefficients are calculated using the LS method. Finally, by using the equalisation in Eq. 7.29 on the resulting matrices, the AGCD of g_1

and g_2 is determined, which is the PSF.

Method 2: The rank estimation process is performed first, for each univariate polynomial of two distorted images, by considering their rows and their columns. The coefficients of AGCD polynomial are then calculated based on the most frequent estimated rank.

To conclude, the algorithm proposed in method 2 suggests a stable solution for the BID problem that does not require *a priori* knowledge of the PSF and the noise.

7.4 Summary

This chapter has described two methods of restoring a PSF, using the method of AGCD computation. The first method identifies the separable PSF, and the second method identifies an arbitrary form of PSF. The separable PSF can be derived given one degraded image; furthermore it can be derived given two degraded images. This method is simple and computationally inexpensive. However, it gives poor results if a nonseparable PSF is convolved with an exact image.
To solve this problem, the HPC is used to compute AGCDs of two degraded images; this leads to the PSF identification and includes the separable and nonseparable forms. In addition, two techniques for the computation of AGCDs have been examined and compared (method 1 and method 2). The first method computes the degree and its coefficients for each iteration. The second method refines the solution by considering the best computed degree. This is determined by selecting the most calculated degree, before proceeding with the computation

of AGCD polynomials. The experimental results show that method 2 yields the best results, compared to method 1.

Since the PSF has been determined, the next step is image deconvolution. The restored image can be obtained by deconvolving the calculated PSF with the degraded image. The method of deconvolution is introduced in the next chapter.

Chapter 8

Image deconvolution

8.1 Introduction

The earlier chapters have described the convolution of a true image and a PSF in two cases: in the presence of noise and in its absence. The convolution is carried on using Toeplitz matrix, where the zero boundary condition has been assumed. In particular, the 2D convolution of a true image with a 2D PSF uses the structured matrix BTTB, which is defined in Section 2.4. It has been shown that the PSF can be determined using the GCD computation which relies on the Sylvester resultant matrix and its subresultant matrices. This chapter describes the reverse procedure of the convolution, more precisely, deconvolution operation. The deconvolution is required for image deblurring, where it is a Toeplitz least squares problem. The use of a structured matrix BTTB is computationally expensive. Section 8.2 considers two methods for the deconvolution that are described in Section 8.2.1 and Section 8.2.2. Section 8.3 discusses some specifications for image deconvolution. Section 8.4 presents some results of image deconvolution.

8.2 Deconvolution

This section considers the polynomial division in 1D and 2D problems. The deconvolution is achieved by dividing the polynomial $g(x, y)$ of the degraded image $\mathcal{G}$ by the polynomial $p(x, y)$ of the computed PSF $\mathcal{P}$, such that

$$\tilde{F} = G/P, \quad \text{or} \quad \tilde{f}(x, y) = g(x, y)/p(x, y), \tag{8.1}$$

where $\tilde{f}(x, y)$ and $\tilde{F}$ are the reconstructed polynomial and its corresponding matrix form, respectively, of a restored image $\tilde{\mathcal{F}}$. The problem in Eq. 8.1 can be solved in 1D using the Toeplitz matrix, if the PSF is separable; moreover, it can be solved using a fast Fourier transform (FFT) if the PSF has an arbitrary form. The following sections consider the deconvolution of images (i.e. 2D polynomials) that have been blurred with separable and nonseparable PSFs.

8.2.1 Deconvolution with separable PSF

Consider a matrix $F \in \mathbb{R}^{M \times N}$ of $\mathcal{F}$, and a matrix $P \in \mathbb{R}^{C \times D}$ of a separable PSF, where $C < M$ and $N < D$. The convolution of F and P yield a matrix $G \in \mathbb{R}^{M+C-1 \times N+D-1}$. Matrix P is given by

$$P = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,D-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,D-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{C-1,0} & p_{C-1,1} & \cdots & p_{C-1,D-1} \end{bmatrix}.$$

Also, matrix P can be expressed as

$$P = \mathbf{P}_c \mathbf{P}_r^T, \tag{8.2}$$

where $\mathbf{P}_c$ and $\mathbf{P}_r$ are the decomposed column and row vectors of matrix P. The polynomial forms of $\mathbf{P}_c$ and $\mathbf{P}_r$ are $p_c(x)$ and $p_r(y)$ respectively, such that

$$p_c(x) = \sum_{k=0}^{d} p_{d-k} x^{d-k}, \quad \text{and} \quad p_r(y) = \sum_{l=0}^{c} p_{c-l} y^{c-l}, \tag{8.3}$$

where $d = D - 1$ and $c = C - 1$. Let H_c and H_r be two Toeplitz matrices, such that

$$H_c = H_c\big(p_c(x)\big) \in \mathbb{R}^{(M+C-1)\times(M)}, \quad \text{and} \quad H_r = H_r\big(p_r(y)\big) \in \mathbb{R}^{(N+D-1)\times(N)};$$

$$H_c = \begin{bmatrix} c_c & & \\ c_{c-1} & \ddots & \\ \vdots & \ddots & c_c \\ c_0 & \vdots & c_{c-1} \\ & \ddots & \vdots \\ & & c_0 \end{bmatrix}, \quad H_r = \begin{bmatrix} r_d & & \\ r_{d-1} & \ddots & \\ \vdots & \ddots & r_d \\ r_0 & \vdots & r_{d-1} \\ & \ddots & \vdots \\ & & r_0 \end{bmatrix}.$$

It follows that

$$R_1 = [H_c]^\dagger, \quad \text{and} \quad R_2 = [H_r]^\dagger, \tag{8.4}$$

where $\dagger$ denotes pseudo-inverse, and R_1 and R_2 are the inverse of the Toeplitz matrices H_c and H_r respectively.

The solution of the inverse problem in Eq. 8.1 is obtained by deconvolving each row of matrix G with the inverse Toeplitz matrix R_2, which contains the information of the horizontal components of the PSF. This procedure can be described as below:

$$\begin{aligned} A(0,n) &= R_1(0,n) * [G(0,n)]^T, \\ A(1,n) &= R_1(1,n) * [G(1,n)]^T, \\ A(2,n) &= R_1(2,n) * [G(2,n)]^T, \\ &\vdots \\ A(M+C-1,n) &= R_1(M+C-1,n) * [G(M+C-1,n)]^T, \end{aligned}$$

for $n = 0, 1, \cdots, N$, and where $*$ denotes matrix multiplication.
The resulting matrix is $A(m,n)$, for $m = 0, 1, \cdots, M+C-1$ and $n = 0, 1, \cdots, N$. Image $\tilde{\mathcal{F}}$ is obtained from the deconvolution procedure of the transposed matrix $[A(m,n)]^T$ and R_2, which stores the information of the vertical components of the PSF P. This procedure can be describes as below:

$$\begin{aligned} B(m,0) &= R_2(m,0) * [A(m,0)]^T, \\ B(m,1) &= R_2(m,1) * [A(m,1)]^T, \\ B(m,2) &= R_2(m,2) * [A(m,2)]^T, \\ &\vdots \\ B(m,N+D-1) &= R_2(m,N+D-1) * [A(m,N+D-1)]^T, \end{aligned}$$

for $m = 0, 1, \cdots, M$.

Finally, the resulting matrix $B(m, n)$ for $m = 0, 1, \cdots, M$ and $n = 0, 1, \cdots, N$, represent the restored image $\tilde{\mathcal{F}}$.

The next section solves the image deconvolution problem when the PSF is not separable.

8.2.2 Deconvolution with nonseparable PSF

The technique described in Section 8.2.1 solves the deconvolution problem when the PSF is separable – more precisely, it has a rank one. In this section, another technique is considered to deconvolve the PSF that has a rank larger than one – more precisely, when it is nonseparable.

It was seen earlier that the 2D convolution problem is given by $\mathbf{g} = H\mathbf{f} + \mathbf{n}$. Here, matrix H is ill-conditioned, and $\mathbf{f}, \mathbf{g}$ and $\mathbf{n}$ are the column vectors of matrices F, G and N respectively. This problem can be solved using the structured matrices which were introduced in Section 2.5.1. The inverse problem is given by $\tilde{f} = H^{-1}\mathbf{g}$. Matrix H is defined as Toeplitz for a 1D separable PSF, and it is defined as BTTB for a 2D nonseparable PSF.

Many deconvolution problems implement the Kronecker product when the PSF has a rank one, and its approximation when the PSF has a rank larger than one [38]. Although the computation of this problem in the spatial domain preserves images information, it tends to be costly. Given a distorted image $\mathcal{G} \in \mathbb{R}^{(M+C-1)\times(N+D-1)}$, and if the estimated 2D PSF is of dimension $(C-1 \times D-1)$, the deconvolved image $\mathcal{F}$ would be of order $M \times N$. A restored image $\tilde{\mathcal{F}}$ is ob-

tained using the polynomial division defined in Eq. 8.1, by performing a fast algorithm based on 2D DFT.

The 2D Fourier transform of image $\mathcal{G}$ is expressed in bivariate polynomial as:

$$g(z_1, z_2) = \frac{1}{\text{(M+C-1)}} \frac{1}{\text{(N+D-1)}} \sum_{s=0}^{\text{(M+C-2)}} \sum_{t=0}^{\text{(N+D-2)}} g(s,t) z_1^s z_2^t,$$

where

$$z_1 = \exp\Big(- 2\pi i \frac{m}{\text{(M+C-1)}} \Big), \quad \text{and} \quad z_2 = \exp\Big(- 2\pi i \frac{n}{\text{(N+D-1)}} \Big), \tag{8.5}$$

on the unit circle in the complex plane, for the counters m and n, such that $m = 0, \cdots, M + C - 1$ and $n = 0, \cdots, N + D - 1$.

Two assumptions are made when using the DFT. First, that the periodic boundary condition holds for the input image $\mathcal{G}$. Second, the discrete points of the PSF are assumed to be the same as the discrete points of the input image $\mathcal{G}$.

Therefore the PSF is extended to be of order $(M+C-1)\times(N+D-1)$ by padding matrix P of $\mathcal{P}$ with zeros. Then the 2D Fourier transform of $\mathcal{P}$ is expressed as a 2D polynomial as:

$$p(z_1, z_2) = \frac{1}{\text{(M+C-1)}} \frac{1}{\text{(N+D-1)}} \sum_{s=0}^{\text{(M+C-2)}} \sum_{t=0}^{\text{(N+D-2)}} p(s,t) z_1^s z_2^t,$$

on the unit circle in the complex plane Eq. 8.5, for the counters m and n, such that $m = 0, \cdots, M + C - 1$ and $n = 0, \cdots, N + D - 1$.

The transformation from the spatial domain into a Fourier domain is achieved using the MATLAB function **fft2**. A point-wise division operation of the 2D

polynomials $g(z_1, z_2)$ and $p(z_1, z_2)$ is performed, such that

$$\tilde{F} = G \oslash P, \quad \text{and} \quad \tilde{f}(z_1, z_2) = g(z_1, z_2)/p(z_1, z_2), \tag{8.6}$$

where $\oslash$ denotes the element-wise division operation, and the matrices F, G and P are expressed in z_1 and z_2 variables.

Matrix $\tilde{F}$ represents the restored image in the Fourier domain. An inverse procedure of 2D FFT is applied to $\tilde{f}(z_1, z_2)$, to transform it into the spatial domain. This procedure is achieved using the MATLAB function `ifft2`. The resulting 2D polynomial $\tilde{f}(x_1, x_2)$ is expressed in x and y variables, and represents the restored image in the spatial domain.

The next section considers some computational specifications implemented in this research.

8.3 Computational specifications

This section discuses some of the main computational details for the implementation of the BID using AGCD computation. The PSF and image selection are discussed in Section 8.3.1, noise addition in Section 8.3.2, and boundary conditions in Section 8.3.3.

8.3.1 The PSF and image selections

It was shown above that the blurred image results from the convolution of the true image with the PSF. In most real applications, the PSF is unknown, and thus the deconvolution requires an estimate of the exact PSF. Therefore, in this

work it is assumed that the PSF is unknown; in particular, that it can take any form including an arbitrary PSF or those described in Section 2.3. Here, the determination of the PSF covers the separable type that has a rank one, and the nonseparable type that has a rank larger than one. Recall from Section 2.3 that the PSF is scaled such that the sum of its entries is always one. It should also be noted that the PSF has smaller dimensions than the exact image. The convolution of the PSF and the true image is performed using the MATLAB function `conv2`. The generated images are chosen such that they contain a number of details (i.e. the images are busy).

8.3.2 Noise addition

The noise in the tested images is added such that it is uniformly distributed in the interval $[-1, \cdots, 1]$. The distorted images are contaminated with two types of noise: the measurement error and the additive noise. One is added into the PSF, while the other is added after the convolution of the true image with the noisy PSF.

8.3.3 Boundary conditions

In general cases, the boundary conditions of a blurred image are not known, as was discussed in Chapter 2. It is important to define the boundary condition for an image before the convolution with the PSF. Failure to do this will lead to incorrect results. The zero condition is assumed throughout this work. The distorted images, which are used in the investigation, are obtained assuming the model 4.1, and the distortion process is illustrated in Algorithm 5.

Algorithm 5 Distorion process

Input

- Two true images $\mathcal{F}_1$ and $\mathcal{F}_2$ of dimensions $M \times N$.
- A PSF $\mathcal{P}$.
- The boundary condition to be imposed (zero).
- The SNR ϵ^{-1} that includes, measurement error $\mathcal{E}$, and noise $\mathcal{N}$.

Output Distorted versions $\mathcal{G}_1$ and $\mathcal{G}_2$

.

Begin

1. Assigned both images $\mathcal{F}_1$ and $\mathcal{F}_2$ into larger images, and define the boundary condition to zero.
2. Add noise $\mathcal{E}_1$ into $\mathcal{P}$ resulting in $\mathcal{G}_1$, and $\mathcal{E}_2$ into $\mathcal{P}$ resulting in $\mathcal{G}_2$.
3. Perform the calculation

$$\mathcal{G}_1 = \mathcal{F}_1 \otimes \big(\mathcal{P} + \mathcal{E}1\big), \quad \text{and} \quad \mathcal{G}_2 = \mathcal{F}_2 \otimes \big(\mathcal{P} + \mathcal{E}2\big).$$

4. Add noise $\mathcal{N}_1$ and $\mathcal{N}_2$ into each image resulted from Step 3, and thereby obtaining the distorted images $\mathcal{G}_1$ and $\mathcal{G}_2$, such that each has the form in Eq. 4.1.

The next section provides a brief discussion of the parallel computation using the sun grid engine (SGE).

8.3.4 High performance algorithm

This section considers the use of networked HPC to solve the BID problem. The iterative computation of an AGCD using the Sylvester matrix approach, discussed

in Section 7.3.2, is highly expensive. Therefore the computations of AGCDs are performed using distributed networked environments. Given two distorted images formed by considering what has been mentioned in Sections 8.3.1, 8.3.2, and 8.3.3, the aim is to calculate the PSF based on parallel computations of AGCDs. The computation of AGCDs of two distorted images, carried out by processing their rows and their columns, is more efficient using distributed networked environments as they provide more resources. The resources include processors (CPU), memory, and storage space, which all reduce the computation time. The parallel algorithm deals with each computation of an AGCD of two inexact univariate polynomials as a task. Then the task is sent to the HPC server with a specified memory and allocation. This step is applied on $(M + C - 1)$ AGCDs of univariate polynomials for each row of the distorted images; this is repeated with columns, a further $(N + D - 1)$ AGCDs computations.

The University of Sheffield uses the Iceberg server for HPC, where each user is assigned a limited number of running tasks in parallel, while the remaining tasks will be queued. However, the computations of AGCDs are performed faster in the Iceberg server than on a user desktop. It follows that another procedure is required to collect the obtained results and calculate the estimated PSF.

8.3.5 Performance measurement

Computational complexity The complexity ($\mathcal{O}$) of this algorithm mostly depends on the GCD computation steps. The input of the implemented GCD is two polynomials, represented as rows or columns, both of the same size

$n-1$. The residual method is used to compute the degree of an AGCD, based on the QR factorisation. This factorisation has a complexity of $\mathcal{O}(n^3)$ flops, which requires 3 operations, which consist of deleting a row and two columns ($3mn$), for each Sylvester matrix [31].

If the QR factorisation is applied for each subresultant matrix, then the complexity will be increased to $\mathcal{O}(n^4)$ flops [31]. While, the QR updating method preforms the QR decomposition on the first Sylvester matrix, and then updates the results from the previous iteration without performing the QR decomposition again. Using the QR updating method decreases the complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$. Thus, the complexity of the iterative AGCD computations will be $\mathcal{O}(n^{3pq})$, where p and q are the total numbers of all rows and all columns respectively [18]. Through the use of parallelisation the complexity of the algorithm is decreased to $\mathcal{O}(n^3)$.

Comparisons As mentioned above, some quantities are used to measure the noise level in the exact, distorted, and restored images. One such measure is SNR, which is expressed as

$$\text{SNR} = \frac{P_{groundtruthsignal}}{P_{noise}},$$

where P denotes the average power of the ground-truth signal that represents a clean image, and the noise signal that represents the distortion in an image. The SNR was defined in Eq. 1.1 using dB units. Similarly, peak signal-to-noise ratio (PSNR) is commonly used to measure the noise level

in an image. The PSNR can be defined in dB units as

$$\text{PSNR}_{db} = 20 * \log_{10}\big(\frac{\text{MAX}}{\sqrt{\text{MSE}}}\big),$$

where MAX is the maximum pixel value contained in the image (i.e. greatest value in the ground-truth signal), and MSE is the mean square error between the ground-truth signal and the distorted or estimated signal, which can be defined as below:

$$\text{MSE} = \frac{\Sigma(\hat{f}(i,j) - f(i,j))^2}{M \times N},$$

where $\hat{f}$ and f are the true images to be compared with, such that both has size $M \times N$.

The RMSE and NAE are used here to measure the performance of the implemented algorithm, such that

$$\text{RMSE} = \frac{\sqrt{\Sigma(\hat{f}(i,j) - f(i,j))^2}}{M \times N}, \quad \text{and} \quad \text{NAE} = \frac{\sum |(\hat{f}(i,j) - f(i,j))|}{\sum |\hat{f}(i,j)|}.$$

The next section discuses some experimental results of the image deconvolution described in this chapter, including separable and nonseparable PSF. However, more experimental results and comparisons with other image deblurring methods will be detailed in the next chapter.

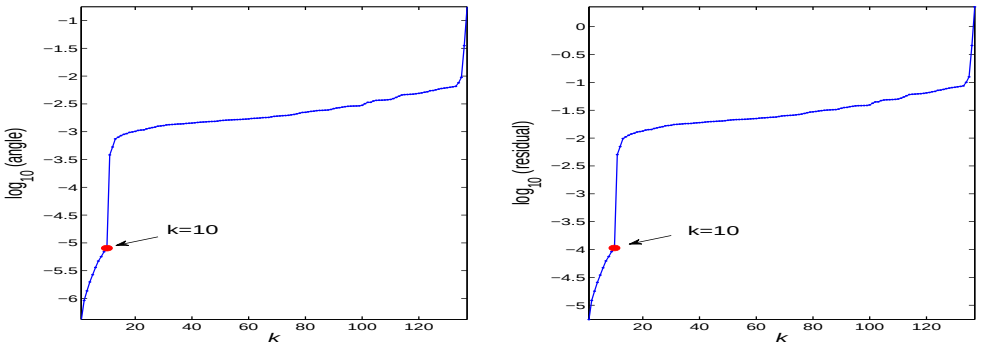
8.4 Examples

This section presents four examples of the implemented BID. Examples 8.1 and 8.2 consider the image restoration when the PSF is separable. Examples 8.3 and 8.4 provide an illustration of the image restoration when the PSF is not separable.

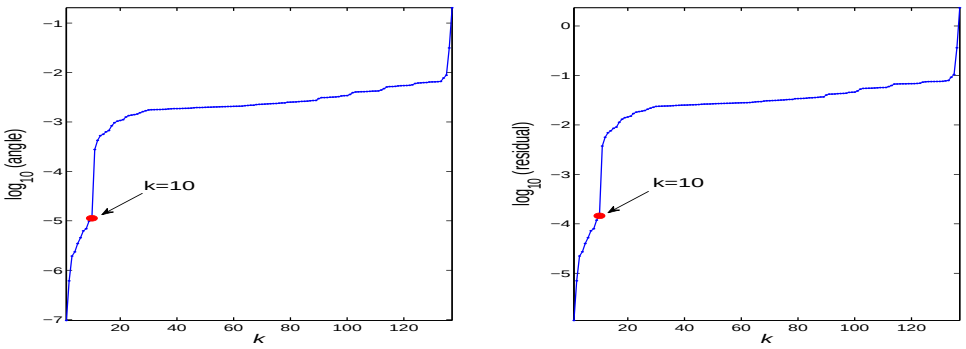
Example 8.1. Consider an exact image of size 128×128 pixels that is convolved with a separable PSF of size 11×11 pixels and the errors $\mathcal{E} = 1e-3$ and $\mathcal{N} = 1e-4$, such that the SNR = 0.6536dB. The exact and distorted images can be seen in Figure 8.3 (a) and (b). To estimate the convolved PSF, two AGCD computations are performed on the degraded image, by considering two rows and two columns. Figure 8.4, demonstrates the obtained ranks for each AGCD computation, which suggests that the estimated PSF is of dimension 11 in both the x and y directions. It can be noted that the residual and the first principal angle methods show similar degree. The entries of the estimated PSF are calculated based on the residual method and the SNTLN output, such that the coefficients are calculated by the LS method. Figure 8.2 illustrates the exact PSF and the estimated PSF, where the error between the exact and the computed PSF is $(8.6146e-04)$. The exact, degraded, and restored images are illustrated in Figure 8.3. The restored image is obtained using polynomial division:

First by deconvolving each row of the degraded image with the computed row vector, which represents the horizontal components of the computed PSF.

Second by deconvolving each column in the matrix that resulted from the first step, with the computed column vector, which represents the vertical com-

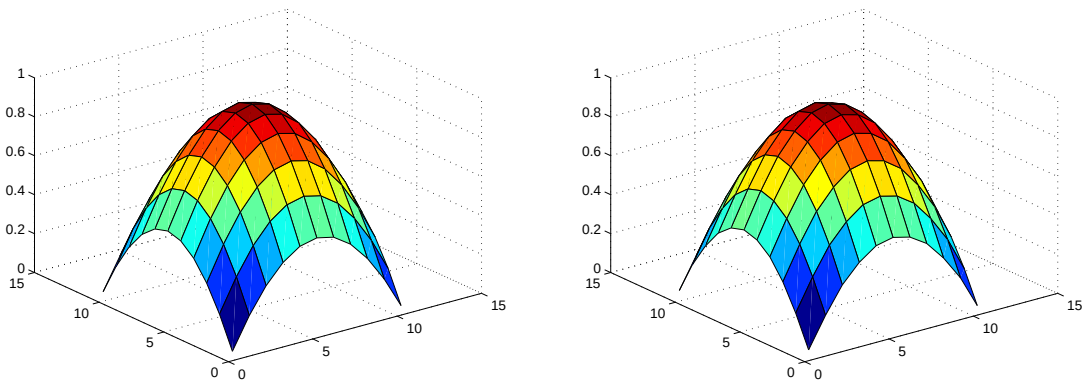


(a) The rank in column direction



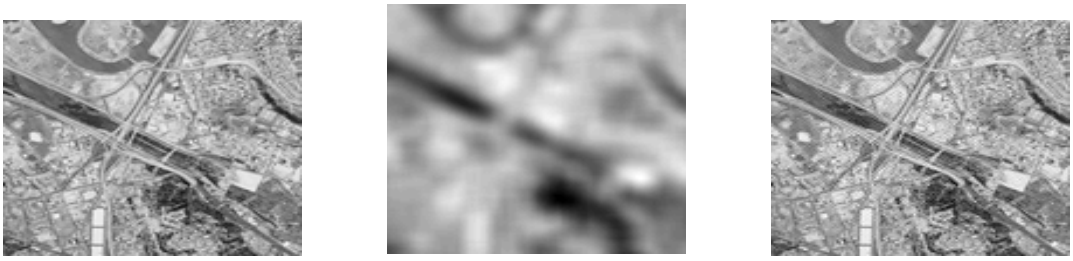
(b) The rank in row direction

Figure 8.1: *The rank estimation based on residual and first principal angle of two inexact polynomials, for Example 8.1*: (a) in column direction, (b) in row direction.



(a) Exact PSF **(b)** Estimated PSF

Figure 8.2: *The exact and restored PSF, for Example 8.1.*



(a) Exact image **(b)** Distorted image **(c)** Restored image

Figure 8.3: *Image restoration using the described method, for Example 8.1.*

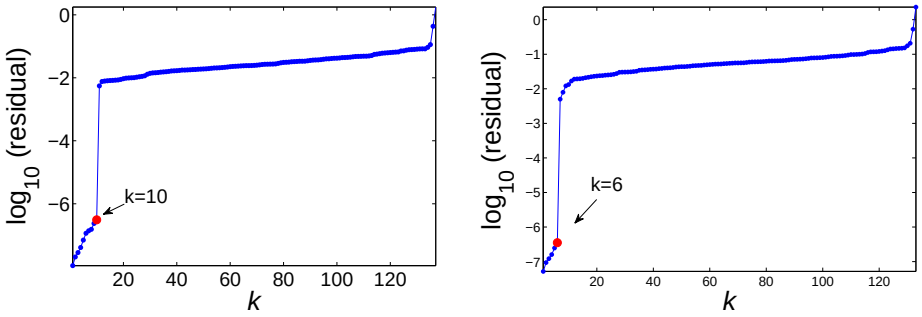
Table 8.1: The quality measurements for Example 8.1

Image	Measurement	Degraded	Restored	Restored
Aerial	SNR	$6.53316e-01$	$1.73112e+01$	Restored
	PSNR	$3.63886e+01$	$7.56052e+01$	Restored
	RMSE	$1.26655e-03$	$1.78256e-04$	Restored
	NAE	$2.18448e-01$	$2.85492e-02$	Restored

ponents of the computed PSF.

Table 8.1 describes the quality measures of the exact, the distorted, and the restored images.

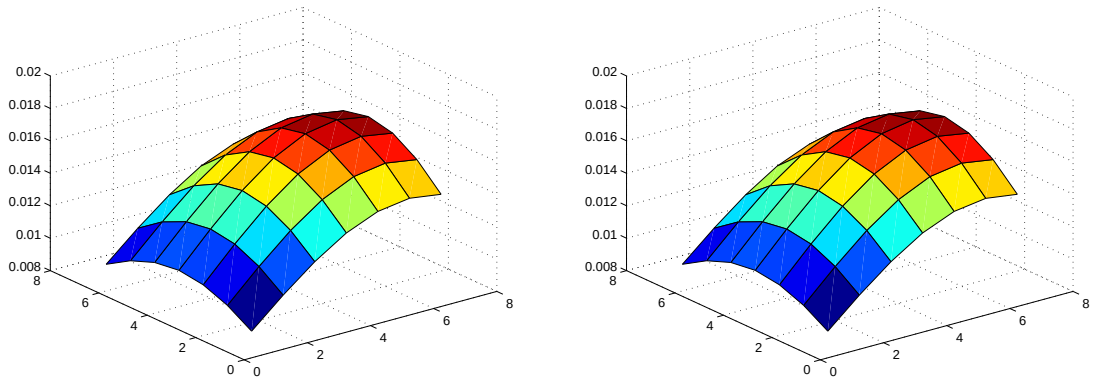
Example 8.2. This experiment is performed on two exact images, each of which is of size 128×128 pixels. Each exact image is convolved with a separable PSF of size 7×11 pixels, and the errors $\mathcal{E}$ and $\mathcal{N}$ provided in Table 8.2. To determinate the convolved separable PSF from the distorted images, two AGCD computations are used, by considering a row and a column from each image. The calculated ranks are shown in Figure 8.4 using the residual method and based on the QR decomposition. The obtained degree of the AGCD polynomial of two rows is 10, and that of two columns is 6. Thus the estimated PSF of size 7×11. Figure 8.5



(a) The rank in column direction **(b)** The rank in row direction

Figure 8.4: *The rank estimation based on residual of two inexact polynomials for Example 8.2:* (a) in column direction, (b) in row direction.

shows the exact and the estimated PSF, and the error in the computed AGCD is $(5.7118e-06)$. The exact, blurred and restored images are shown in Figures 8.6



(a) Exact PSF **(b)** Estimated PSF

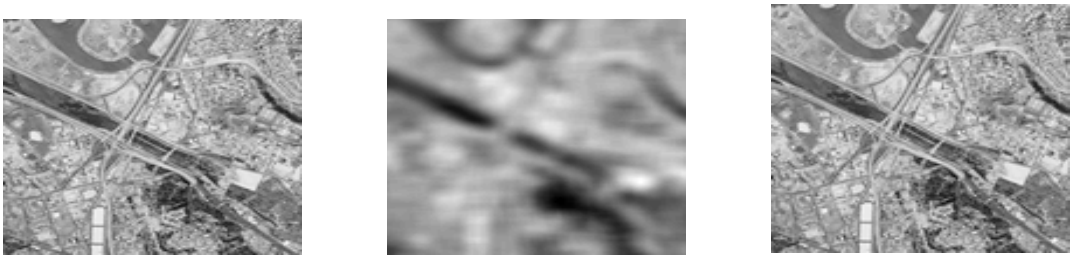
Figure 8.5: *The exact and restored PSF for Example 8.2.*

and 8.7. It can be seen that the method presented in this chapter yields improved deblurred images, as shown in Table 8.2.

Example 8.3. This example considers the methods of AGCD that have been described in this chapter. Consider two exact images, each of size 128×128

(a) Exact image **(b)** Distorted image **(c)** Restored image

Figure 8.6: *Image restoration for Grass−image in Example 8.2.*



(a) Exact image **(b)** Distorted image **(c)** Restored image

Figure 8.7: *Image restoration for Aerial−image in Example 8.2.*

pixels. These images are convolved with an arbitrary PSF of size 9×7 pixels, the measurement errors $\mathcal{E}$, and noise $\mathcal{N}$, such that the blurring model in Eq. 4.1 is assumed for each image. Thus the distorted images are constructed based on the information provided in Table 8.3. The convolved PSF here is generated such that its rank is larger than one, and it is scaled such that its entries sum to one. The true images and their degraded forms are shown in Figure 8.8, such that the degraded *Rice*-image has SNR = 2.78 and the *Blobs*-image has SNR = 3.80.

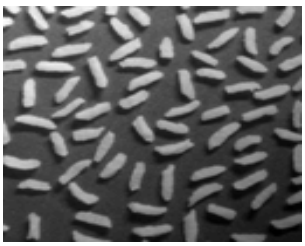
Consider the degraded versions of *Rice*-image and *Blobs*-image, with no prior knowledge of the convolved PSF or the noise levels. The implementation of an AGCD algorithm on both images, using Method 1 as described in this chapter,

Table 8.2: The quality measurements of the restored images in Example 8.2

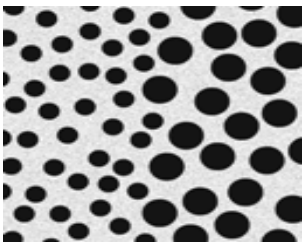
Image	E	N	Measurement	Degraded	Restored
Grass	$1.0000e-05$	$1.0000e-10$	SNR	$-1.33956e+00$	$7.67426e+01$
			PSNR	$3.36665e+01$	$2.15830e+02$
			RMSE	$1.45121e-03$	$1.60731e-07$
			NAE	$2.26932e-01$	$2.42103e-05$
Aerial	$1.0000e-09$	$1.0000e-11$	SNR	$1.32079e+00$	$8.82165e+01$
			PSNR	$3.97401e+01$	$2.36667e+02$
			RMSE	$1.07114e-03$	$5.67050e-08$
			NAE	$1.76023e-01$	$9.70245e-06$

Table 8.3: Test data information for Example 8.3.

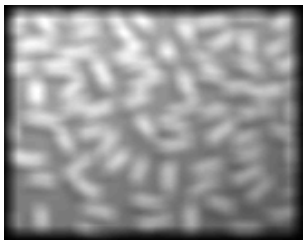
Image	E	N	SNR	PSNR	RMSE	NAE
rice	$1e-06$	$1e-03$	$2.78331e+00$	$2.72609e+01$	$1.99906e-03$	$6.28097e-01$
blobs	$1e-05$	$1e-07$	$3.80258e+00$	$2.63399e+01$	$2.09327e-03$	$3.56309e-01$



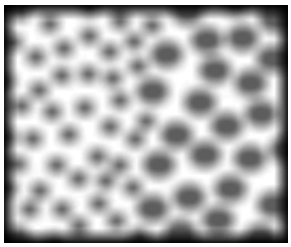
(a) Clean *Rice*-image



(b) Clean *Blobs*-image

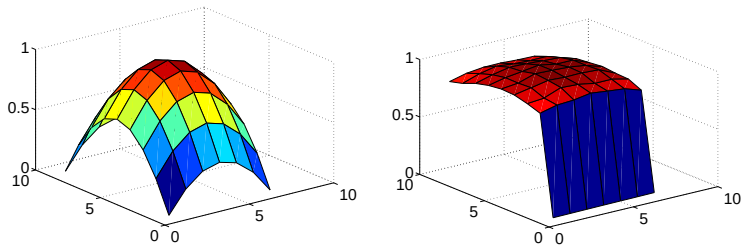


(c) Distorted *Rice*-image

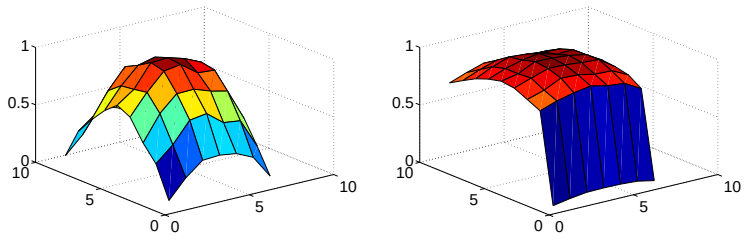


(d) Distorted *Blobs*-image

Figure 8.8: *Test images, for Example 8.3*: (a) and (b) are the original images, and their degraded versions (c) and (d) respectively.



(a) Exact **(b)** Estimated 1

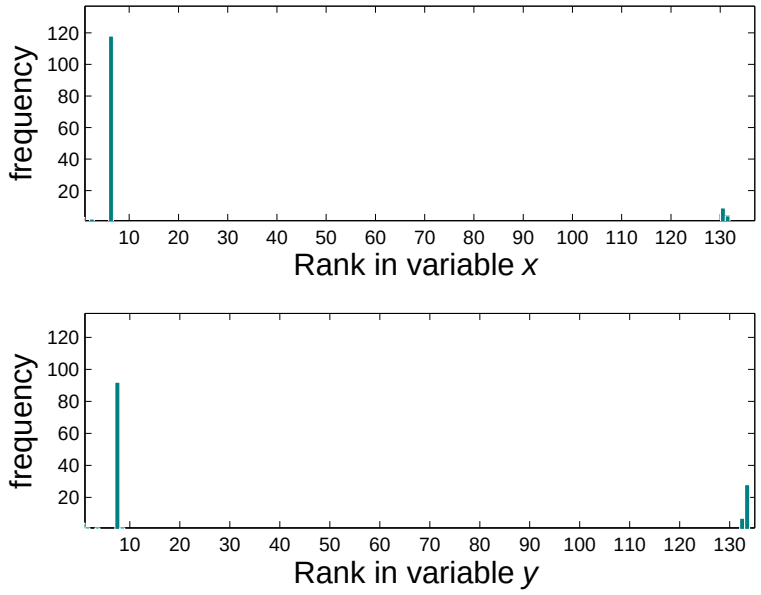


(c) Estimated 2 **(d)** Average of 1 and 2

Figure 8.9: *The exact and estimated PSFs, using Method* 1 *of AGCD computations, for Example 8.3*: (a) is the exact PSF, (b) Estimated 1: is the PSF calculated by considering the rows, resulted in $e_1 = 0.3099$, (c) Estimated 2: is the PSF calculated by considering the columns, resulted in $e_2 = 0.0078$, and (d) is the average of the estimated PSFs, resulted in $e_3 = 0.1557$.

yields two estimations of the PSF, as shown in Figure 8.9. One is the result of using the AGCD algorithm on each row of the degraded images, that is the scaled $d(m, n)$ in Eq. 7.24, and the other is formed by considering the columns, that is the scaled $d(m, n)$ in Eq. 7.25. It can be seen that, although the AGCD algorithm gives a good result in one computation, as shown in (c) with error $e_2 = 0.00078$, it gives a poor result in the other one, as shown (b) with error $e_1 = 0.3099$, and thus applying Eq. 7.29 results in (d) with error $e_3 = 0.1557$.

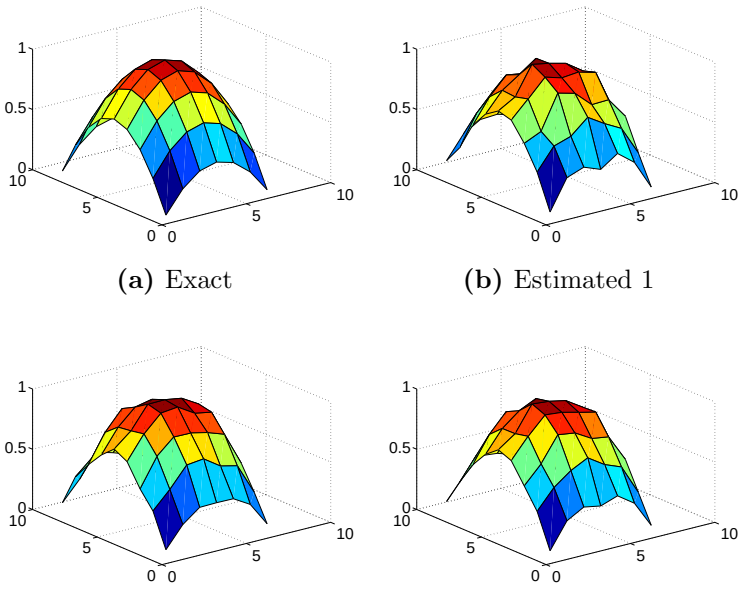
The computation of the AGCD is performed on each univariate polynomial that corresponds to each row in *Rice*-image and *Blobs*-image. Figure 8.10 (a) shows that the most frequent computed rank, in this case, is $\tilde{d}_1 = 8$. Figure 8.10



(a)

Figure 8.10: *Rank estimation for Example 8.3*

(b) shows that the most frequent computed rank is $\tilde{d}_2 = 6$, by considering the columns of *Rice*-image and *Blobs*-image. Thus the degree of the estimated PSF equals 9 in the x direction and 7 in the y direction. It was pointed out above that the computation method of AGCDs, presented in Algorithm 4, may yield better results if the coefficients are calculated based on the most frequent computed rank. To examine the difference between Method 1 and Method 2, let the input data be exactly the same as given in Table 8.3. The rank estimation of two degraded images gives $\tilde{d}_1 = 8$ and $\tilde{d}_2 = 6$ as the most frequent computed ranks with respect to variables x and y, respectively. Figure 8.11, shows a significant improvement in the estimated PSF as the output of Eq. 7.29, with an error of (0.0075). Consequently, the restored images are improved too, as shown in Figure 8.4. The restored images are then obtained using polynomial division of



(a) Exact **(b)** Estimated 1

(c) Estimated 2 **(d)** Average of 1 and 2

Figure 8.11: *The exact and estimated PSFs, using Method* 2 *of AGCD computations for Example 8.3*: (a) is the exact PSF, (b) Estimated 1: is the PSF calculated by considering the rows, resulted in $e_1 = 0.0097$, (c) Estimated 2: is the PSF calculated by considering the columns, resulted in $e_2 = 0.0073$, and (d) is the average of the estimated PSFs resulted in $e_3 = 0.0075$.

their degraded versions and the computed PSF, such that the computed PSF is the one resulting from Eq. 7.29. Figure 8.12 demonstrates images restoration using Method 1 in (a) and (b), and Method 2 in (c) and (d). The performance

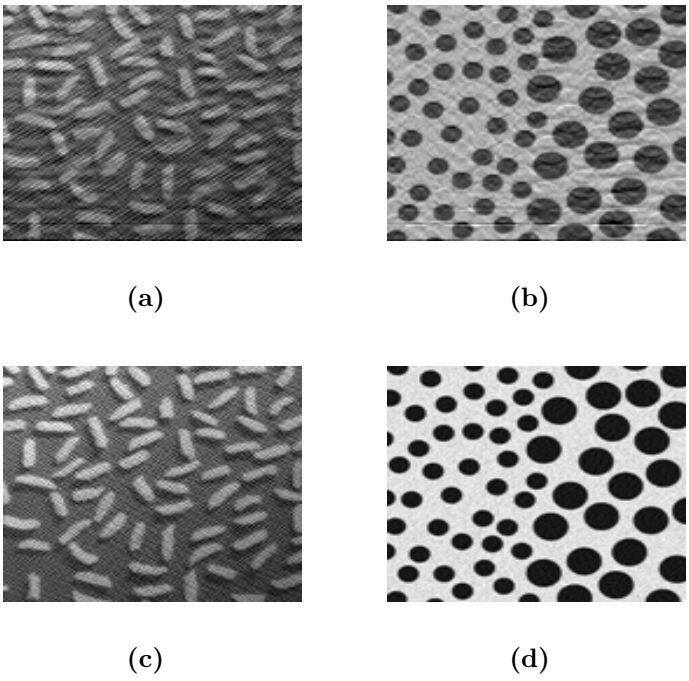


Figure 8.12: *Image restoration using the computed PSF, where (a) and (b) are restored based on Method* 1*, and (c) and (d) are restored based on Method* 2 *for Example 8.3.*

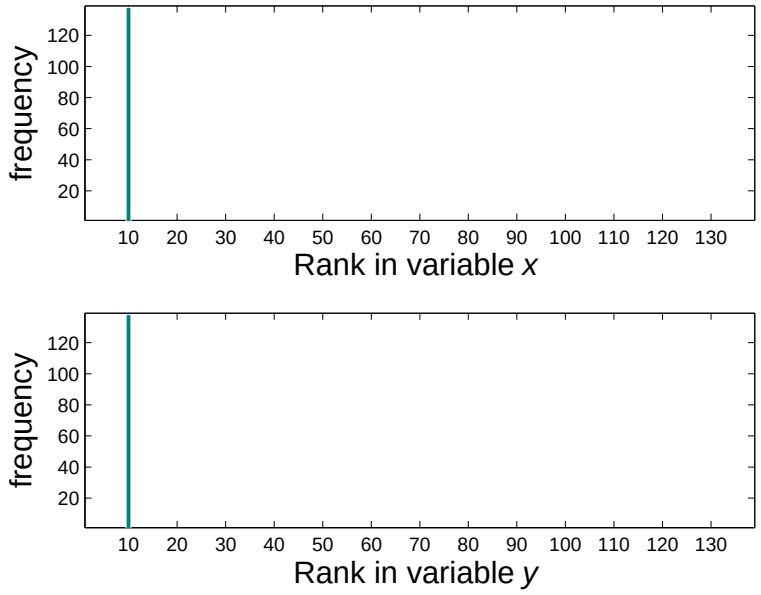
measurements for the restored images are shown in Table 8.4. In general, Method 2 provides more accurate results compared to Method 1, and therefore Method 2 is the best way to solve the BID problem.

Example 8.4. This experiment is performed using Method 2 on two exact images, each of size 128×128 pixels. The degraded images are formed using a PSF of size 11×11 pixels and errors $\mathcal{E}$ and $\mathcal{N}$ as shown in Table 8.5. The estimated degree is 11 in both variable x and y, as shown in Figure 8.13, because the most frequent

Table 8.4: The quality measures of the restored images in Example 8.3

Image	Measurement	Method 1	Method 2
rice	SNR	$4.23633e+00$	$8.27808e+00$
	PSNR	$4.15697e+01$	$4.69995e+01$
	RMSE	$9.77499e-04$	$7.45091e-04$
	NAE	$2.82914e-01$	$2.21285e-01$
blobs	SNR	$5.48644e+00$	$2.40934e+01$
	PSNR	$3.10015e+01$	$7.42875e+01$
	RMSE	$1.65806e-03$	$1.90396e-04$
	NAE	$3.38758e-01$	$3.46211e-02$

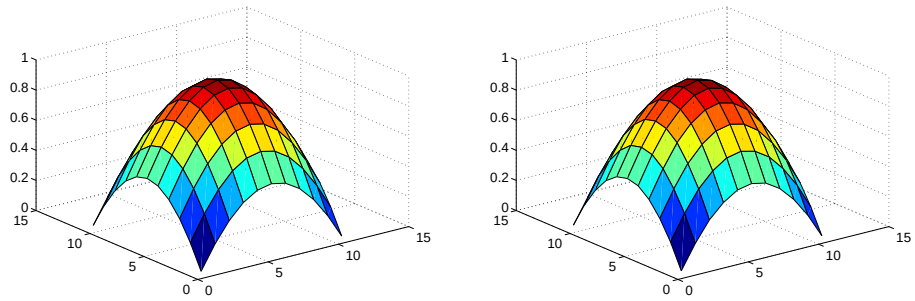
calculated rank is 10 in variables x and y. Subsequently, the PSF is estimated



(a)

Figure 8.13: *The rank estimation for Example 8.4*

using LS, based on the computed degree. The exact and the estimated PSF are illustrated in Figure 8.14, and the error in the computed AGCD is $(4.0762e-04)$. In particular, the error resulting by considering the rows of the two distorted images in the estimated PSF is $(3.7736e-04)$, and is $(3.4754e-04)$ by con-



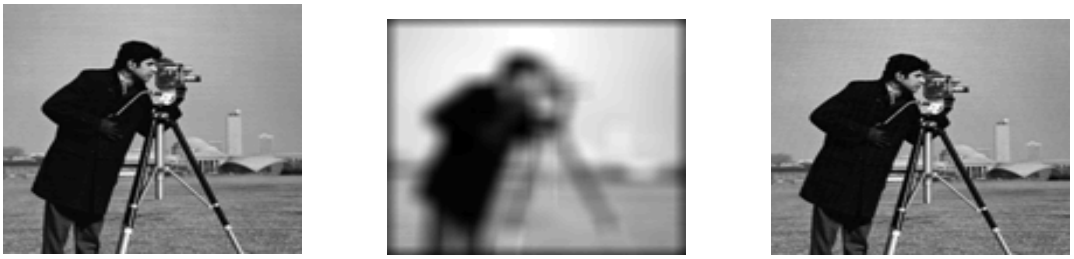
(a) Exact **(b)** Estimated

Figure 8.14: *The exact and estimated PSFs, using Method* 2 *of AGCD computations, for Example 8.4*: (a) is the exact PSF, (b) is the estimated PSF with error $(4.0762e-04)$.

sidering their columns. This means that the AGCD algorithm has achieved the best result with the column-wise computation. However, the average of row-wise computation and column-wise computation is considered for image restoration. It has been found that the resulting errors in the estimated PSFs, using the three methods LS, LU and QR are equal, at $(4.0762e-04)$. Figures 8.15 and 8.16 demonstrate the result of image restoration for the two degraded images, with no prior knowledge of the PSF or the noise level. Table 8.5, shows a notable

(a) Exact **(b)** Distorted **(c)** Restored

Figure 8.15: *Image restoration, using the computed PSF from Method* 2*, for Example 8.4.*



(a) Exact **(b)** distorted **(c)** Restored

Figure 8.16: *Image restoration, using the computed PSF from Method* 2*, for Example 8.4.*

improvement in the quality of the restored images.

Table 8.5: The quality measurements of the restored images in Example 8.4

Image	E	N	Measurement	Degraded	Restored
Lena	$1.0000e-03$	$1.0000e-07$	SNR	$4.05850e+00$	$2.67272e+01$
			PSNR	$3.95120e+01$	$9.25314e+01$
			RMSE	$1.08342e-03$	$7.64710e-05$
			NAE	$2.30603e-01$	$1.71584e-02$
Aerial	$1.0000e-04$	$1.0000e-08$	SNR	$4.17403e+00$	$3.44451e+01$
			PSNR	$3.33368e+01$	$1.06735e+02$
			RMSE	$1.47533e-03$	$3.75903e-05$
			NAE	$3.39274e-01$	$8.28852e-03$

8.5 Summary

This chapter has described two methods for image deconvolution, after which the PSF in estimated. One method considered the separable PSF and the other considered the nonseparable PSF. The first method uses two Toeplitz matrices for the deconvolution with the degraded image, where each Toeplitz is formed by a 1D PSF. This accelerates the solution of the deconvolution problem. The second method uses the Fourier transform for the deconvolution, implementing a 2D FFT

computation. Image restoration is achieved in two steps: first, by determining the PSF, then by deconvolving the degraded image with the computed PSF. The experimental results have been evaluated using the SNR, PSNR, RMSE, and NAE measures.

Results from the BID methods that are described in this thesis will be compared with state-of-the-art techniques in Chapter 9.

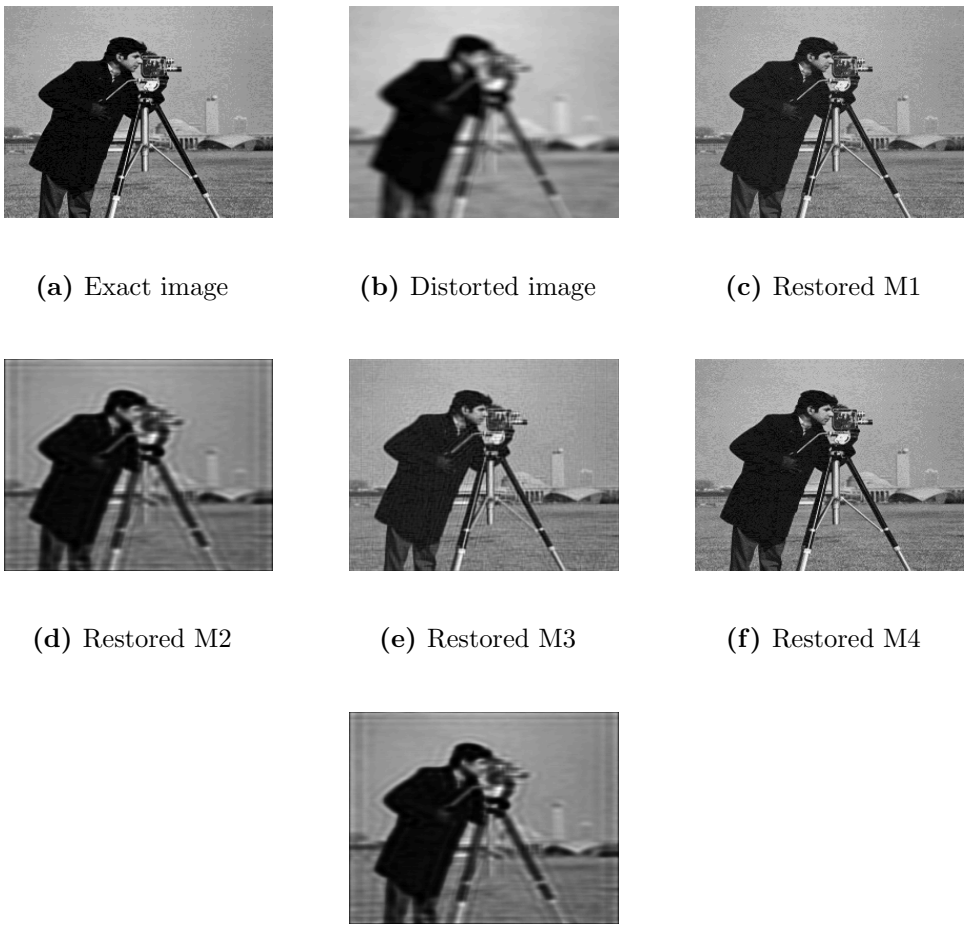
Chapter 9

Experimental results

The implementation of an AGCD computation in the BID problem was considered in Chapter 7. The performance of the described method can be seen in Example 7.1, where the degraded images have been improved to some extent. This case, however, is appropriate with a separable form of PSF. Chapter 7 has solved the BID problem for an arbitrary form of PSF. The implemented solution of the BID problem can be seen in Examples 8.1, 8.2, 8.3, and 8.4.

This chapter analyses more examples for both cases, where solving the BID problem can restore a distorted based on separable, or nonseparable PSF. All results are compared with the state-of-the-art methods, such as the Wiener filter, regularised filter, the Lucy-Richardson method, and blind maximum likelihood deconvolution, and where each has built-in MATLAB function. The state-of-the-art methods were discussed in Chapter 2, where it was noted that all of them require the knowledge *a priori* of the PSF or the noise. The threshold arguments in the state-of-the-art methods include the PSF and noise level. The work in this thesis does not require prior knowledge of the PSF and noise level, and thus it has

an advantage over the other methods. The noise level should be omitted; to be more precise, it will be specified to zero in the MATLAB functions: `deconvwnr`, `deconvreg`, `deconvlucy`, and `deconvblind`.



(a) Exact image **(b)** Distorted image **(c)** Restored M1

(d) Restored M2 **(e)** Restored M3 **(f)** Restored M4

(g) Restored M5

Figure 9.1: *Image restoration using five deblurring methods of Cameraman−image presented in Example 9.1*: (a) The original image; (b) the distorted image; (c) the proposed method in this thesis; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

Example 9.1. Consider an exact image $\mathcal{F}$ of size 265×265 pixels, distorted by

a PSF of size 7×11, $\mathcal{E} = 1e-03$, and $\mathcal{N} = 1e-04$, such that

$$\begin{aligned} \text{SNR} &= 7.37779e+00, \quad \text{PSNR} = 4.27513e+01, \\ \text{NAE} &= 2.05333e-01, \quad \text{RMSE} = 4.60709e-04. \end{aligned}$$

Figure 9.1 shows the exact, degraded, and restored images.

Table 9.1: Comparison of five deblurring methods. M1: The proposed method in this thesis, M2: Lucy-Richardson, M3: Regularised filter, M4:Wiener filter, M5: Maximum likelihood.

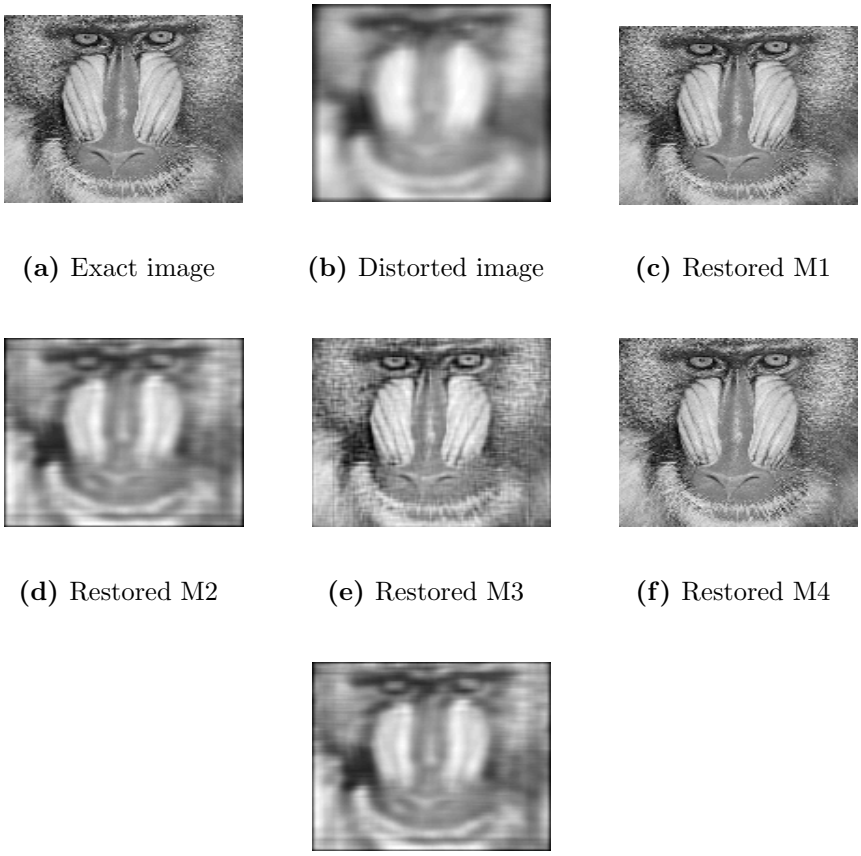
Image	*Methods*	PSF	SNR	PSNR	RMSE	NAE
Cameraman	M1	Computed	$1.71819e+01$	$6.42544e+01$	$1.57215e-04$	$6.42144e-02$
	M2	Exact	$9.10841e+00$	$4.50260e+01$	$4.11181e-04$	$1.73433e-01$
	M3	Exact	$2.05577e+01$	$7.22362e+01$	$1.05480e-04$	$4.57103e-02$
	M4	Exact	$1.38809e+01$	$5.95191e+01$	$1.99214e-04$	$8.19056e-02$
	M5	Exact	$9.16524e+00$	$4.54500e+01$	$4.02555e-04$	$1.68222e-01$

Example 9.2. Consider two degraded images of two dissimilar scenes, distorted by a separable 7×11 PSF, and the errors $\mathcal{E}$ and $\mathcal{N}$, as shown in Table 9.2, and where each exact image is of order 128×128.

Table 9.2: Test data information.

Image	E	N	SNR	PSNR	RMSE	NAE
Mandrill	$1e-03$	$1e-07$	$2.12109e+00$	$3.80893e+01$	$1.16330e-03$	$2.21922e-01$
Cameraman	$1e-04$	$1e-06$	$4.81392e+00$	$3.38244e+01$	$1.43980e-03$	$3.34378e-01$

The original, degraded, and restored images of the mandrill and the cameraman are illustrated in Figures 9.2, and 9.3 respectively.



(a) Exact image **(b)** Distorted image **(c)** Restored M1

(d) Restored M2 **(e)** Restored M3 **(f)** Restored M4

(g) Restored M5

Figure 9.2: *Image restoration using five deblurring methods of Mandril−image presented in Example 9.2*: (a) the original image; (b) the distorted image; (c) the proposed method in this thesis; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

(a) Exact image

(b) Distorted image

(c) Restored M1

(d) Restored M2

(e) Restored M3

(f) Restored M4

(g) Restored M5

Figure 9.3: *Image restoration using five deblurring methods of Cameraman−image presented in Example 9.2*: (a) the original image; (b) the distorted image; (c) the proposed method in this thesis; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

The obtained images using the state-of-the-art methods with the method described in this thesis are compared, with regard to the quantity measures SNR, PSNR, RMSE, and NAE. Table 9.3 shows that the proposed method gives the

best results of the five deblurring methods.

Table 9.3: Comparison of five deblurring methods. M1: The proposed method in this thesis, M2: Lucy-Richardson, M3: Regularised filter, M4: Wiener filter, M5: Maximum likelihood.

Image	*Methods*	PSF	SNR	PSNR	RMSE	NAE
Mandrill	M1	Computed	$3.74384e+01$	$1.14291e+02$	$2.57629e-05$	$5.04868e-03$
	M2	Exact	$3.21622e+00$	$4.14797e+01$	$9.81908e-04$	$1.77973e-01$
	M3	Exact	$7.15808e+00$	$4.94026e+01$	$6.60732e-04$	$1.23529e-01$
	M4	Exact	$2.46838e+01$	$8.13193e+01$	$1.33956e-04$	$2.70837e-02$
	M5	Exact	$3.23316e+00$	$4.14918e+01$	$9.81310e-04$	$1.78080e-01$
Cameraman	M1	Computed	$5.48765e+01$	$1.49276e+02$	$4.48032e-06$	$9.62942e-04$
	M2	Exact	$8.17104e+00$	$4.73201e+01$	$7.33240e-04$	$1.25952e-01$
	M3	Exact	$1.49994e+01$	$6.05686e+01$	$3.78058e-04$	$7.59597e-02$
	M4	Exact	$3.39301e+01$	$9.87412e+01$	$5.60598e-05$	$1.25656e-02$
	M5	Exact	$8.21684e+00$	$4.74460e+01$	$7.28640e-04$	$1.24617e-01$

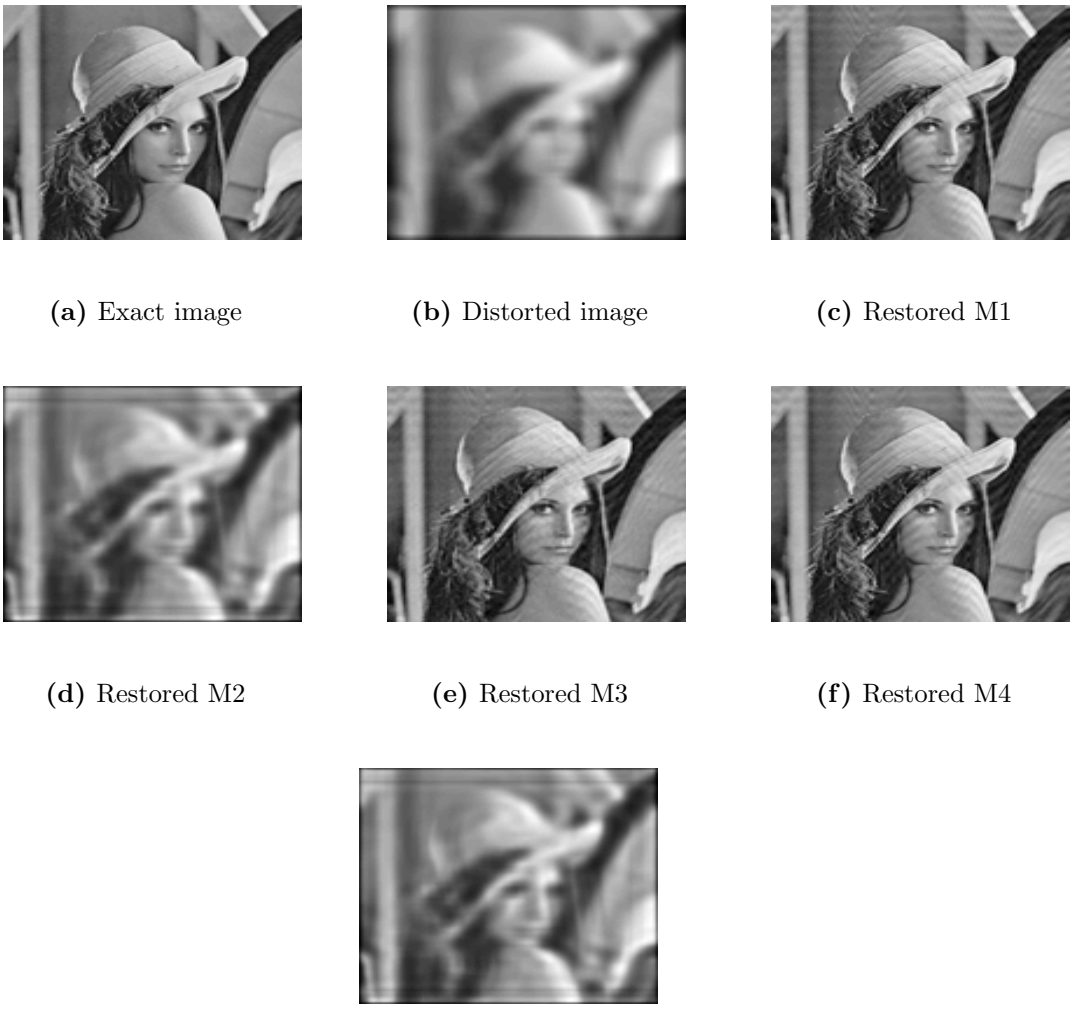
Example 9.3. This example considers two dissimilar images, such that each is a degraded version of an exact image of size 128×128. Table 9.4 describes the errors $\mathcal{E}$ in a nonseparable PSF of size 7×11 pixels, and additive noise $\mathcal{N}$ that are used to generate the test images.

Table 9.4: Test data information.

Image	E	N	SNR	PSNR	RMSE	NAE
Lena	$1e-04$	$1e-07$	$1.89393e+01$	$7.46394e+01$	$1.87075e-04$	$3.81702e-02$
Cameraman	$1e-05$	$1e-06$	$1.86431e+01$	$6.39916e+01$	$3.18589e-04$	$7.34911e-02$

The restored images obtained using the described Method 1 in Chapter 7 are compared with the other methods in Figures 9.4 and 9.5. Table 9.5 describes the quality measures for each *Lena*-image and *Cameraman*-image. It shows that the

best results are obtained using the proposed method. The restored images using the proposed method and the Wiener filter give comparable results in *Lena*-image, while the proposed method works much better for the *Cameraman*-image.



(a) Exact image **(b)** Distorted image **(c)** Restored M1

(d) Restored M2 **(e)** Restored M3 **(f)** Restored M4

(g) Restored M5

Figure 9.4: *Image restoration using five deblurring methods of Lena−image presented in Example 9.3*: (a) the original image; (b) the distorted image; (c) the proposed method using Method 1 as described in Chapter 7; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

(a) Exact image

(b) Distorted image

(c) Restored M1

(d) Restored M2

(e) Restored M3

(f) Restored M4

(g) Restored M5

Figure 9.5: *Image restoration using five deblurring methods of Cameraman−image presented in Example 9.3*: (a) the original image; (b) the distorted image; (c) the proposed method using Method 1 as described in Chapter 7; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

Table 9.5: Comparison of five deblurring methods. M1: The proposed method in this thesis, M2: Lucy-Richardson, M3: Regularised filter, M4:Wiener filter, M5: Maximum likelihood.

Image	*Methods*	PSF	SNR	PSNR	RMSE	NAE
Lena	M1	Computed	$1.89393e+01$	$7.46394e+01$	$1.87075e-04$	$3.81702e-02$
	M2	Exact	$6.68150e+00$	$4.14797e+01$	$8.19420e-04$	$1.77973e-01$
	M3	Exact	$1.77809e+01$	$7.22342e+01$	$2.10981e-04$	$4.28863e-02$
	M4	Exact	$1.89349e+01$	$7.46187e+01$	$1.87269e-04$	$3.82255e-02$
	M5	Exact	$6.77892e+00$	$4.50161e+01$	$8.22770e-04$	$1.69682e-01$
Cameraman	M1	Computed	$4.55410e+01$	$1.26863e+02$	$1.37404e-05$	$3.06153e-03$
	M2	Exact	$8.01989e+00$	$4.69717e+01$	$7.46128e-04$	$1.30980e-01$
	M3	Exact	$1.64757e+01$	$6.45442e+01$	$3.09907e-04$	$6.02402e-02$
	M4	Exact	$1.86431e+01$	$6.39916e+01$	$3.18589e-04$	$7.34911e-02$
	M5	Exact	$8.15652e+00$	$4.72390e+01$	$7.36221e-04$	$1.29591e-01$

Example 9.4. This experiment compares the obtained results of Method 2 in Example 8.4 with the state-of-the-art methods. Example 8.4 demonstrated the success of using Method 2 over Method 1 when each assumes the arbitrary form of the PSF and uses the HPC to calculate the AGCDs of two polynomials. The exact images are of size 128×128 pixels, where each convolved by a PSF of size 11×11 pixels. The errors in the *Lena*-image are $\mathcal{E} = 1e-03$ and $\mathcal{N} = 1e-07$, which result in SNR $= 4.05850 + 00$, and the errors in *Cameraman*-image are $\mathcal{E} = 1e-04$ and $\mathcal{N} = 1e-08$, which result in SNR $= 4.17403e+00$.

It can be seen that there is a remarkable improvement in the restored images, based on Figures 8.15 and 8.16, and Table 8.5; however, it is important to compare these outputs with the other methods. Table 9.6 provides a comparison of the proposed solution of the BID problem that uses Method 2 described in Chapter 7 with four deblurring methods: Lucy-Richardson, regularised filter, Wiener

filter, and maximum likelihood.

Table 9.6: Comparison of five deblurring methods, for Example 9.4. M1: Method 2 as described in Chapter 7, M2: Lucy-Richardson, M3: Regularised filter, M4:Wiener filter, M5: Maximum likelihood.

Image	*Methods*	PSF	SNR	PSNR	RMSE	NAE
Lena	M1	Computed	$3.04157e+01$	$1.01318e+02$	$4.92819e-05$	$1.07073e-02$
	M2	Exact	$6.04064e+00$	$4.38625e+01$	$8.71621e-04$	$1.76771e-01$
	M3	Exact	$1.93994e+01$	$7.59614e+01$	$1.75110e-04$	$3.63449e-02$
	M4	Exact	$2.67272e+01$	$9.25314e+01$	$7.64710e-05$	$1.71584e-02$
	M5	Exact	$6.14273e+00$	$4.37759e+01$	$8.75403e-04$	$1.79228e-01$
Cameraman	M1	Computed	$3.44451e+01$	$1.06735e+02$	$3.75903e-05$	$8.28852e-03$
	M2	Exact	$6.49265e+00$	$4.13813e+01$	$9.86750e-04$	$2.11048e-01$
	M3	Exact	$1.59412e+01$	$5.58216e+01$	$6.98177e-05$	$1.10849e-01$
	M4	Exact	$3.40906e+01$	$9.43519e+01$	$1.37404e-05$	$1.66366e-02$
	M5	Exact	$6.62351e+00$	$4.13600e+01$	$9.87799e-04$	$2.14263e-01$

(a) Exact image

(b) Distorted image

(c) Restored M1

(d) Restored M2

(e) Restored M3

(f) Restored M4

(g) Restored M5

Figure 9.6: *Image restoration using five deblurring methods of Lena−image presented in Example 9.4*: (a) the original image; (b) the distorted image; (c) the proposed method using Method 2 as described in Chapter 7; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

(a) Exact image

(b) Distorted image

(c) Restored M1

(d) Restored M2

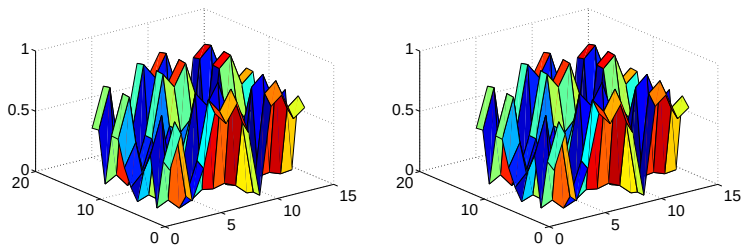
(e) Restored M3

(f) Restored M4

(g) Restored M5

Figure 9.7: *Image restoration using five deblurring methods of Cameraman−image presented in Example 9.4*: (a) the original image; (b) the distorted image; (c) the proposed method using Method 2 as described in Chapter 7; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

Example 9.5. Consider an exact image $\mathcal{F}$ of order 200×240 pixels, distorted by a PSF of size 13×13 pixels, and the errors $\mathcal{E} = (1e-04)$ and $\mathcal{N} = (1e-06)$. Therefore a degraded image $\mathcal{G}$ of $\mathcal{F}$ is obtained with SNR $= (-2.63651e-01)$, PSNR $= (2.39979e+01)$, RMSE $= (1.37490e-03)$, and NAE $= (1.18966e+00)$. The method described in Chapter 7 is applied to restore an image from its degraded version with no prior information of the PSF and noise level, and the results are shown in Figure 9.8.

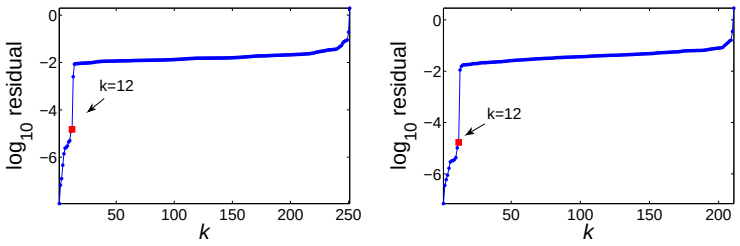


(a) Exact PSF **(b)** Estimated PSF

Figure 9.8: *The exact and estimated PSF.*

In particular, the PSF in Figure 9.8 is obtained using two AGCD computations of two polynomials, one by considering two rows of $\mathcal{G}$ and the other by considering two columns of $\mathcal{G}$. These polynomials are normalised and preprocessed, such that two modified Sylvester matrices are obtained, each of the form $S(\bar{g}_1(\theta_0, w), \alpha_0 \bar{g}_2(\theta_0, w))$.

The results show that $\alpha_0 = 1.0773$ and $\theta_0 = 1.0028$ in the row-wise, and $\alpha_0 = 0.9337$ and $\theta_0 = 1.0004$ in the column-wise computation. The degree of an AGCD of two inexact polynomials is equal to the rank loss of the modified Sylvester matrix of two inexact polynomials. Thus the estimated degrees in the two AGCD

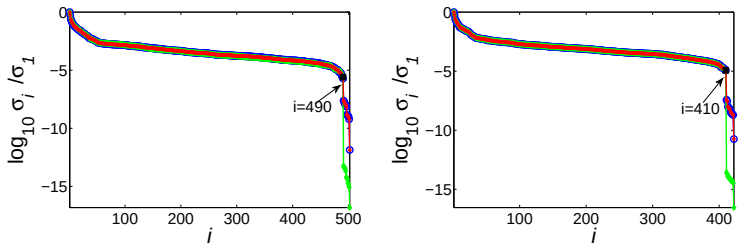


(a) in column direction **(b)** in row direction

Figure 9.9: *The rank estimation based on the residual and QR updating methods.*

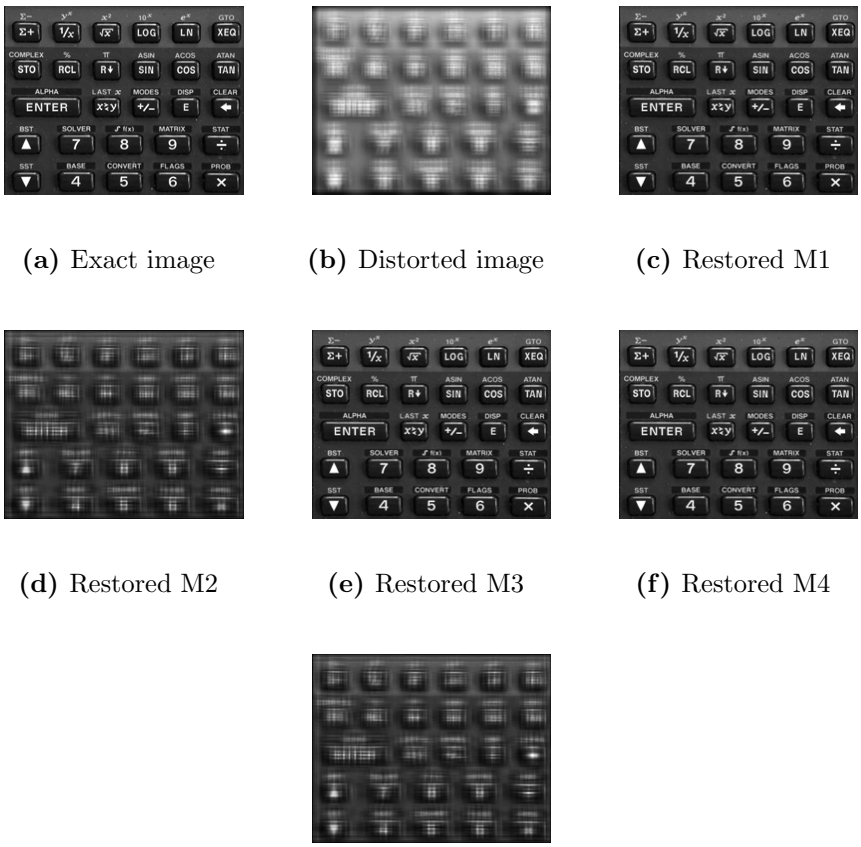
computations is 13 in both in column and row directions, using the residual method based on the QR decomposition tool for rank estimation, as shown in Figure 9.9.

This was followed by performing two SNTLN computations on the polynomials of two rows and likewise on the polynomials of two columns from $\mathcal{G}$, in order to calculate corrected forms of each. Figure 9.10 shows the normalised singular values of Sylvester matrices $S(g_1(x), \alpha g_2(x))$, $S(\bar{g}_1(\theta, w), \alpha \bar{g}_2(\theta, w))$, and $S(\tilde{g}_1(w), \alpha \tilde{g}_2(w))$. It can be seen that the rank of the Sylvester matrices of the two polynomials is equal to 410 in column direction, and 490 in row direction.



(a) in column direction **(b)** in row direction

Figure 9.10: *The normalized singular values of* $S(\bar{g}_1(\theta, w), \alpha \bar{g}_2(\theta, w))$ $*$, $S(g_1(x), \alpha g_2(x))$ $\circ$, *and* $S(\tilde{g}_1(w), \alpha \tilde{g}_2(w))$ $\Diamond$.



(a) Exact image **(b)** Distorted image **(c)** Restored M1

(d) Restored M2 **(e)** Restored M3 **(f)** Restored M4

(g) Restored M5

Figure 9.11: *Image restoration using five deblurring methods of Calculator−image presented in Example 9.5*: (a) the original image; (b) the distorted image; (c) the proposed method using Method 2 as described in Chapter 7; (d) Lucy-Richardson; (e) regularised filter; (f) Wiener filter; (g) maximum likelihood.

Subsequently, the coefficients of the PSF were computed using the corrected forms of the two polynomials in the modified Sylvester matrices, and which are the input of the LS method. The output from these procedures is two AGCD polynomials, where one represents the horizontal components and the other represents the vertical components of the PSF, and their multiplication forms the

estimated PSF. It can be seen that the estimated PSF is an approximation of the exact PSF, and the error in the GCD is $(4.022e-05)$.

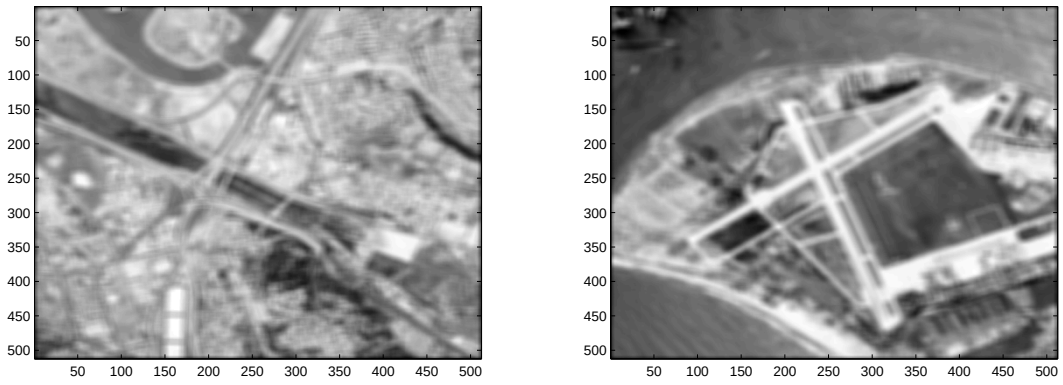
The deconvolution of $\mathcal{G}$ with the computed PSF, using polynomial division, yields a restored image with SNR $= (6.29876e+01)$. Figure 9.11 shows the image restoration results of the five algorithms, using the computed PSF and without the prior knowledge of the noise level. It can be noticed from this figure, that the restored images in (c), (e), and (f) are visually similar in their appearance, but they are different in their noise level and error measurement.

Table 9.7: Comparison of five deblurring methods. M1: The proposed method, M2: Lucy-Richardson, M3: Regularised filter, M4:Wiener filter, M5: Maximum likelihood.

Image	*Methods*	PSF	SNR	PSNR	RMSE	NAE
Calculator	M1	Computed	$6.29876e+01$	$1.81059e+02$	$5.34235e-07$	$3.45151e-04$
	M2	Computed	$3.92373e+00$	$4.51907e+01$	$4.76514e-04$	$2.92871e-01$
	M3	Computed	$3.78079e+01$	$1.20431e+02$	$1.10725e-05$	$8.11109e-03$
	M4	Computed	$4.08961e+01$	$1.29768e+02$	$6.94235e-06$	$4.85405e-03$
	M5	Computed	$4.00354e+00$	$4.54065e+01$	$4.71401e-04$	$2.85128e-01$

Table 9.7 describes the quality measures for the five deblurring methods: each method deconvolves the degraded image $\mathcal{G}$ with the computed PSF obtained from the AGCD computations. This experiment used the computed PSF, that resulted from AGCD computation, for the comparisons with the other deblurring methods. More precisely, each MATLAB function: `deconvlucy`, `deconvreg`, `deconvwnr`, and `deconvblind` employs the estimated PSF in the restoration process. The method using AGCD computation based on the Sylvester matrix approach yields the best results.

Example 9.6. Given two blurred images $\mathcal{G}_1$ and $\mathcal{G}_2$ as in Figure 9.12, the exact and the estimated PSF are shown in Figure 9.13. Figures 9.15 and 9.16 show the obtained results for each image using MATLAB built-in functions, with known exact PSF.



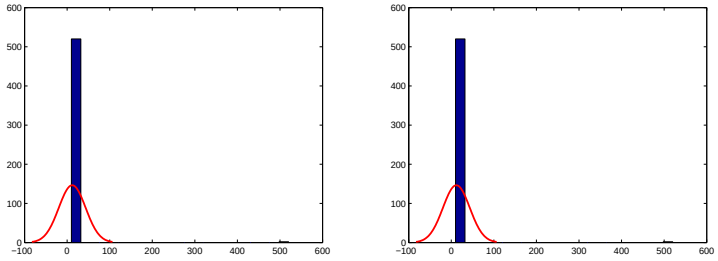
(a) **(b)**

Figure 9.12: *Two different degraded aerial images.*

These images are degraded versions of original images $\mathcal{F}_1$ and $\mathcal{G}_2$, each of order 512×512 and convolved with 11×11 nonseparable PSF. The errors are specified in Table 9.8.

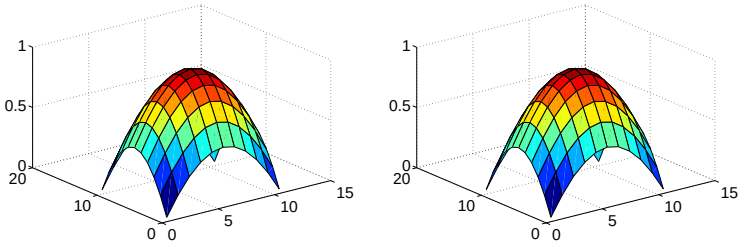
Table 9.8: Test data information in Example 9.6.

Image	E	N	SNR	PSNR	RMSE	NAE
Aerial 1	$1e-04$	$1e-05$	$2.91659e+00$	$3.94656e+01$	$2.71484e-04$	$1.65648e-01$
Aerial 2	$1e-05$	$1e-05$	$6.66705e+00$	$4.58388e+01$	$1.97403e-04$	$1.34142e-01$



(a) **(b)**

Figure 9.14: *The estimated ranks in: (a) x variable, (b) y variable.*



(a) Exact PSF **(b)** Estimated PSF

Figure 9.13: *The exact and estimated PSF.*

The most frequent estimated rank loss are equal to 10, thus the degree in both variables x and y is equal to 11, as shown in the histogram with normal fit in Figure 9.14. Then, the PSF is calculated using the LS method, and the error in the GCD is $(1.5194e-04)$.

(a) Restored M1

(b) Restored M2

(c) Restored M3

(d) Restored M4

(e) Restored M5

Figure 9.15: *Image restoration of Aerial l image using five deblurring methods*: (a) the proposed method using HPC; (b) Lucy-Richardson; (c) regularised filter; (d) Wiener filter; (e) maximum likelihood.

(a) Restored M1

(b) Restored M2

(c) Restored M3

(d) Restored M4

(e) Restored M5

Figure 9.16: *Image restoration of Aerial* 2 *image using five deblurring methods*: (a) the proposed method using HPC; (b) Lucy-Richardson; (c) regularised filter; (d) Wiener filter; (e) maximum likelihood.

Table 9.6 describes the noise level and the errors in the restored images using

the five algorithms.

Table 9.9: Comparison of five deblurring methods. M1: The proposed method, M2: Lucy-Richardson, M3: Regularised filter, M4:Wiener filter, M5: Maximum likelihood.

Image	*Methods*	PSF	SNR	PSNR	RMSE	NAE
Aerial 1	M1	Computed	$2.77919e+01$	$9.44455e+01$	$1.73729e-05$	$1.12771e-02$
	M2	Exact	$4.01077e+00$	$3.80385e+01$	$2.91565e-04$	$1.90565e-01$
	M3	Exact	$9.85029e+00$	$5.30974e+01$	$1.37321e-04$	$8.88698e-02$
	M4	Exact	$1.70739e+01$	$6.79635e+01$	$6.53013e-05$	$4.30298e-02$
	M5	Exact	$4.05273e+00$	$3.81147e+01$	$2.90455e-04$	$1.89930e-01$
Aerial 2	M1	Computed	$3.36956e+01$	$1.06170e+02$	$9.66670e-06$	$7.60212e-03$
	M2	Exact	$8.33640e+00$	$4.84204e+01$	$1.73498e-04$	$1.23611e-01$
	M3	Exact	$1.57198e+01$	$6.67295e+01$	$6.94573e-05$	$5.31436e-02$
	M4	Exact	$1.95935e+01$	$7.44120e+01$	$4.73036e-05$	$3.71912e-02$
	M5	Exact	$8.38941e+00$	$4.85290e+01$	$1.72558e-04$	$1.23084e-01$

It is noted that, the Lucy-Richardson and maximum likelihood deblurring methods produce ringing artifacts in the restored images, while the ringing artifacts is less in the other deblurring methods. It is also noted that, if the Wiener filter has given the exact PSF it produces, in some experiments not all, a comparable results to the proposed method. In general, it is clear that the best results are achieved using the proposed method which is based on the Sylvester matrix, where no prior information of the PSF and noise are given.

The approach described in this work, is able to solve the real world applications, as it will be show in the next example.

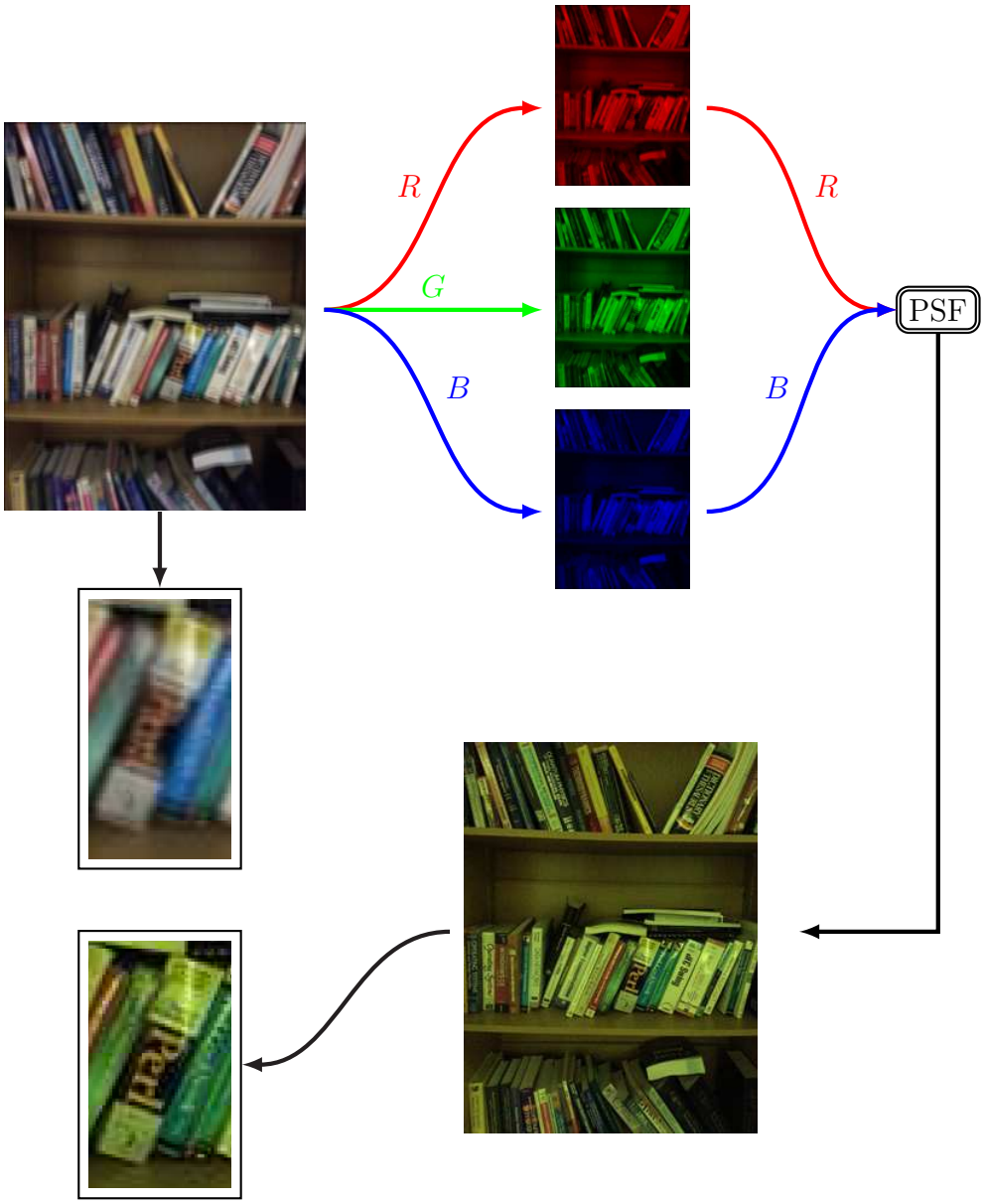


Figure 9.17: Blind deblurring process for one distorted coloured image, where the blur and noise are unknown. R: Red, G: Green, B: Blue.

Example 9.7. Figure 9.17 illustrates the blind deblurring process given one distorted image, which is captured by a smartphone's camera. When the blurring function and the noise are unknown, the problem can be solved using the GCD

algorithm on any two channels of the coloured image (i.e. Red, Green, and Blue), where each contains the same blurring function. In this example the red and blue, are sufficient to estimate the PSF. The restored image is obtained using the polynomial division of the given blurred image and the calculated PSF. It can be seen from the restored image that the word ($Perl$) is clearly recognizable.

9.1 Summary

This chapter has demonstrated the success of image restoration using the Sylvester matrix approach for the AGCD computation of two inexact polynomials compared with the state-of-the-art methods.

Chapter 10

Conclusion and future work

The main work in this thesis has been to present a solution for the BID problem using polynomial computations. It was shown that the BID problem exists in many applications and several methods have been proposed in the literature to solve it. However, these methods require partial knowledge of the blurred image, including the convolved PSF and the noise level. The proposed method in this thesis implements the GCD computation and polynomial division to solve the BID problem, and does not require prior knowledge of the PSF or the noise to reconstruct the exact image from its degraded version.

This problem is solved for the spatially invariant distortion system, where each pixel in the degraded image has the same blur effect, in particular, the same PSF. The separable PSF can be calculated using two GCD computations, by considering two rows and two columns from the blurred image such that each is represented in polynomial forms. In a noise-free environment, the GCD of two exact polynomials is considered and the PSF can be obtained exactly. However, an approximation of the exact PSF can only be obtained when the computation

is performed in a floating point arithmetic; a noise being introduced into the blurring system leads to inexact polynomials. In particular, additive noise leads the exact polynomials to be coprime; thus, an AGCD of two inexact polynomial is considered.

Several algorithms were proposed to define an AGCD of two inexact polynomials; each definition may find different solutions to each problem. In this thesis the degree of an AGCD of two inexact polynomials is considered to be correct when it is equal to the degree of the GCD of their exact polynomials. The AGCD algorithm implemented, based on a method developed in [84], uses the Sylvester resultant matrix and its subresultant matrices, which were defined in Chapter 5. However, three preprocessing operations were performed on the Sylvester matrix to enhance the computations, because of the partitioned nature of the Sylvester matrix of two polynomials.

An AGCD of two inexact polynomials is calculated by determining its degree first. This can be achieved by applying different tools, described in chapter 6, on the modified Sylvester matrix of two inexact polynomials. Two methods were applied to the modified Sylvester resultant matrix and its subresultant matrices to define the degree of an AGCD: the first principle angle and the residual of an approximate linear equation. These methods were studied in [88], and described in Chapter 6, where it was proved that both methods yield similar rank loss with different column index. Using the QR updating technique in the residual method reduces the computation complexity from $\mathcal{O}(n^4)$ flops to $\mathcal{O}(n^3)$ flops.

After the degree of an AGCD was defined the method of SNTLN proposed in [87] was implemented to calculate a minimum perturbation to be added into the Sylvester matrix of two inexact polynomials, such that they have a nonconstant

common factor. The application of this method to the BID problem allows to obtain a much better approximation for the exact PSF. The coefficients of an AGCD polynomial can then be obtained using the LS method on the corrected two inexact polynomials. The experimental results show that using the LS method is more efficient than using the LU or the QR decomposition methods.

The calculated degrees and coefficients construct the dimensions and the entries of the estimated PSF. Similarly, this solution was applied to identify the non-separable PSF by considering each column and row of two degraded images, such that each image has the same PSF. This algorithm has been implemented using the HPC to accelerate the computation of AGCDs. Two techniques to estimate the PSF have been investigated. It was found that using Method 2 yields more accurate results than Method 1. Method 2 applies the majority voting method on the calculated degrees for each AGCD computation, before the coefficients computations, while, Method 1 calculates the degree and its coefficients for each AGCD computation. The exact images were then reconstructed using polynomial division of the degraded images and the estimated PSF; this can be achieved by using a Toeplitz matrix or the FFT method.

The method introduced in this thesis has provided a reliable solution for the BID problem, compared with the state-of-the-art methods as shown in Chapter 9. This method is implemented with the use of parallel computing to solve an arbitrary forms of PSF. The experimental results in this thesis, with different sizes of PSFs and noise levels, have shown the adequacy of the described deblurring method.

The main contributions of this work can be listed as below:

- The 1D and 2D PSF identification, using AGCD computations and with no

prior information of the PSF and noise level.

- Fast deconvolution of a single degraded image and the calculated separable PSF, without using the DFT.
- Fast deconvolution of two different degraded images and the calculated separable PSF, without using the DFT.
- Fast deconvolution of two different degraded images and the calculated arbitrary form of PSF, implemented in parallel network architecture.

One possible improvement for this work is to consider a fast deconvolution algorithm for efficient computation with structured matrices. This algorithm can then be used to deconvolve the PSF array from the blurred image, in the spatial domain. Furthermore, future work can consider the deconvolution of the PSF from the corrected degraded images, which correspond to the corrected inexact polynomials resulted from SNTLN. Another direction for future work is the extension of the described algorithm using AGCD computation, to consider different applications such as video deblurring and spatially variant deblurring. For example, blurs may occur in many frames of a video when the camera is shaking. The proposed approach can be extended into three-dimension (3D) to restore exact frames from distorted ones caused by camera shake.

References

[1] H. Ahn and A.E. Yagle. 2-D blind deconvolution by partitioning into coupled 1-D problems using discrete radon transforms. In *Proceedings International Conference on Image Processing*, volume 2, pages 37–40, October 1995.

[2] M. Almeida and M. Figueiredo. Deconvolving images with unknown boundaries using the alternating direction method of multipliers. *IEEE Transactions on Image Processing*, 22(8):3074–3086, August 2013.

[3] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, March 1997.

[4] S. Barnett. *Polynomials and Linear Control Systems*. Pure and Applied Mathematics. M. Dekker, NY, USA, 1983.

[5] B. Beckermann and G. Labahn. A fast and numerically stable Euclidean–like algorithm for detecting relatively prime numerical polynomials. *Journal of Symbolic Computation*, 26(6):691–714, December 1998.

[6] S. Berisha and J. G. Nagy. Iterative methods for image restoration. Technical report, Mathematics and Computer Science, Emory University, 2013.

[7] J. Biemond, R. L. Lagendijk, and R.M. Mersereau. Iterative methods for image deblurring. *Proceedings of the IEEE*, 78:856–883, May 1990.

[8] D. S. C. Biggs. *Accelerated Iterative Blind Deconvolution*. PhD thesis, Department of Electrical and Electronic Engineering, New Zealand, December 1998.

[9] D. A. Bini and P. Boito. Structured matrix-based methods for polynomial ϵ-GCD: analysis and comparisons. In *Proceedings of ISSAC*, ISSAC '07, pages 9–16, NY, USA, 2007. ACM Press.

[10] C. H. Bischof. Codes for rank-revealing QR factorizations of dense matrices. *ACM Transaction on Mathematical Software*, 24(2):254–257, June 1998.

[11] Å. Björck. *Numerical methods for Least Squares Problems*. SIAM, PA, 1996.

[12] M. Bôcher. *Introduction to Higher Algebra*. The Macmillan, 1957.

[13] R. N. Bracewell. *Two-Dimensional Imaging*. Prentice Hall Signal Processing Series. Prentice hall, 1995.

[14] W. S. Brown. On Euclid's algorithm and the computation of polynomial greatest common divisors. *Journal of the ACM*, 18:478–504, October 1971.

[15] J. B. Campbell. *Introduction to Remote Sensing*. Guilford Press, NY, 2008.

[16] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustics Speech and Signal Processing*, 24(1): 58–63, February 1976.

[17] P. Chin, W. Laurier, and G. F. Corless. Optimization strategies for the approximate GCD problem. In *ISSAC'98*, pages 228–235. ACM Press, 1998.

[18] B. Codenotti and M. Leoncini. *Parallel Complexity of Linear System Solution*. World Scientific, 1991.

[19] G. E. Collins. Subresultants and reduced polynomial remainder sequences. *Journal of the ACM*, 14(1):128–142, January 1967.

[20] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. In *Proceedings of ISSAC*, pages 195–207. ACM Press, 1995.

[21] R. M. Corless, S. M. Watt, and L. Zhi. QR factoring to compute the GCD of univariate approximate polynomials. *IEEE Transactions on Signal Processing*, 52(12):3394–3402, December 2004.

[22] S. Correia, M. Carbillet, A. Richichi, M. Bertero, and P. Boccacci. Large binocular telescope image restoration using simulated adaptively corrected point-spread functions. In *Astronomical Telescopes and Instrumentation*, volume 4006, pages 650–658. International Society for Optics and Photonics, 2000.

[23] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, NY, second edition, 1997.

[24] A. Danelakis, M. Mitrouli, and D. Triantafyllou. Blind image deconvolution using a banded matrix method. *Numerical Algorithms*, 64(1):43–72, September 2013.

[25] B. N. Datta. *Numerical linear algebra and applications*. SIAM, 2010.

[26] I. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. *Journal of Pure and Applied Algebra*, 117 and 118:229–251, May 1997.

[27] I. Z. Emiris, A. Galligo, and H. Lombardi. Numerical univariate polynomial GCD. In *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 323–343. AMS, 1996.

[28] O. Faugeras. *Three-Dimensional computer vision: A geometric viewpoint*. Artificial Intelligence. MIT Press, Cambridge, MA, USA, 1993.

[29] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794, July 2006.

[30] S. Ghaderpanah and S. Klasa. Polynomial scaling. *SIAM Journal on Numerical Analysis*, 27:117–135, February 1990.

[31] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[32] 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, May 1979.

[33] R. C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson/Prentice Hall, 2008.

[34] P. Gravel, G. Beaudoin, and J. A. De Guise. A method for modeling noise in medical images. *IEEE Transactions on Medical Imaging*, 23(10):1221–1232, October 2004.

[35] L. Guan and R. K. Ward. Restoration of randomly blurred images by the wiener filter. *IEEE Transactions on Acoustics Speech and Signal Processing*, 37:589–592, April 1989.

[36] P. C. Hansen. The discrete Picard condition for discrete ill-posed problems. *BIT Numerical Mathematics*, 30(4):658–672, 1990.

[37] P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4):561–580, December 1992.

[38] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. Fundamentals of Algorithms. SIAM, 2006.

[39] M. Hasan. *The computation of multiple roots of a polynomial using structure preserving matrix methods*. PhD thesis, Department of Computer Science, Sheffield, England, 2011.

[40] G. Heinig. Kernel structure of Toeplitz-plus-Hankel matrices. *Linear Algebra and Its Applications*, 340(13):1 – 13, January 2002.

[41] V. Hribernig and H. J. Stetter. Detection and validation of clusters of polynomial zeros. *Journal of Symbolic Computation*, 24:667–681, December 1997.

[42] H. Hudson and T. Lee. Maximum likelihood restoration and choice of smoothing parameter in deconvolution of image data subject to poisson noise. *Computational Statistics and Data Analysis*, 26(4):393–410, 1998.

[43] B. R. Hunt. The application of constrained least squares estimation to image restoration by digital computer. *IEEE Transactions on Computers*, C-22(9): 805–812, September 1973.

[44] H. Kai and M. T. Noda. Hybrid rational function approximation and its accuracy analysis. *Reliable Computing*, 6(4):429–438, February 2000.

[45] E. L. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *Journal of Symbolic Computation*, 47(1):1–15, January 2012.

[46] N. Karmarkar and Y. N. Lakshman. Approximate polynomial greatest common divisors and nearest singular polynomials. In *ISSAC*, pages 35–39, 1996.

[47] N.K. Karmarkar and Y.N. Lakshman. On approximate GCDs of univariate polynomials. *Journal of Symbolic Computation*, 26:653–666, December 1998.

[48] A. K. Katsaggelos. *Digital Image Restoration*. Springer Series in Information Sciences. Springer-Verlag, 1991.

[49] Aggelos K. Katsaggelos and Kuen-Tsair Lay. Maximum likelihood blur identification and image restoration using the EM algorithm. *IEEE Transactions on Signal Processing*, 39(3):729–733, March 1991.

[50] D. Kundur and D. Hatzinakos. A novel recursive filtering method for blind image restoration. In *International Conference on Image and Signal Processing*, pages 428–431, 1995.

[51] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE Signal Processing Magazine*, 13(6):61–63, November 1996.

[52] G. Labahn, D. K. Chio, and S. Cabay. The inverses of block Hankel and block Toeplitz matrices. *SIAM Journal on Computing*, 19(1):98–123, February 1990.

[53] R. L. Lagendijk and J. Biemond. *Iterative Identification and Restoration of Images*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, 1991.

[54] R. L. Lagendijk and J. Biemond. *Basic methods for image restoration and identification*, pages 167–181. Elsevier Academic Press, Burlington, MA, February 2005.

[55] R. L. Lagendijk, J. Biemond, and D.E. Boekee. Identification and restoration of noisy blurred images using the expectation-maximization algorithm. *IEEE Transactions on Acoustics Speech and Signal Processing*, 38(7):1180–1191, July 1990.

[56] R. L. Lagendijk, D. E. Boekee, and J. Biemond. Blur identification using the expectation-maximization algorithm. *IEEE Transactions on Acoustics Speech and Signal Processing*, ASSP-38:1180–1191, July 1990.

[57] K. P. Lee, J. G. Nagy, and L. Perrone. Iterative methods for image restoration: A MATLAB object oriented approach. *Numerical Algorithms*, 36: 73–93, May 2003.

[58] B. Li, Z. Liu, and L. Zhi. A fast algorithm for solving the Sylvester structured total least squares problem. *Signal Processing*, 87(10):2313 – 2319, March 2007.

[59] T. Y. Li and Z. Zeng. A rank-revealing method with updating, downdating and applications. *SIAM Journal on Matrix Analysis and Applications*, 26 (4):918–964, 2005.

[60] Z. Li, Z. Yang, and L. Zhi. Blind image deconvolution via fast approximate GCD. In *Proceedings of the 2010 ISSAC*, ISSAC '10, pages 155–162. ACM Press, July 2010.

[61] R. Liu and J. Jia. Reducing boundary artifacts in image deconvolution. In *15th IEEE International Conference on Image Processing*, pages 505–508, October 2008.

[62] I. Markovsky and S. V. Huffel. Overview of total least-squares methods. *Signal Processing*, 87(10):2283–2302, October 2007.

[63] A. K. Nandi. *Blind Estimation Using Higher-Order Statistics*. Springer, 1999.

[64] M. Noda and T. Sasaki. Approximate GCD and its application to ill-conditioned algebraic equations. In *Proceedings of the International Symposium on Computation Mathematics*, volume 38 of *ISCM '90*, pages 335–351, Amsterdam, The Netherlands, 1991. North-Holland Publishing.

[65] V. Y. Pan. Approximate polynomial GCDs, pade approximation, polynomial zeros and bipartite graphs. In *Proceedings of The 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98, pages 68–77, PA, USA, 1998. SIAM.

[66] V. Y. Pan. Computation of approximate polynomial GCDs and an extension. *Information and Computation*, 167:71–85, June 2001.

[67] M. Petrou and P. Bosdogianni. *Image Processing: The Fundamentals*. Wiley, 1999.

[68] M. Petrou and C. Petrou. *Image Processing: The Fundamentals*. John Wiley & Sons, 2010.

[69] S. U. Pillai and B. Liang. Blind image deconvolution using a robust GCD approach. *IEEE Transactions on Image Processing*, 8(2):295–301, February 1999.

[70] P. Premaratne and C.C. Ko. Retrieval of symmetrical image blur using zero sheets. *IEE Proceedings: Vision, Image and Signal Processing*, 148(1):65–69, February 2001.

[71] S. J. Reeves and R. M. Mersereau. Blur identification by the method of generalized cross-validation. *IEEE Transactions on Image processing*, 1(3): 301–311, July 1992.

[72] W. H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, January 1972.

[73] J. M. Sanches, J. C. Nascimento, and J. S. Marques. Medical image noise reduction using the Sylvester-lyapunov equation. *IEEE Transactions on Image Processing*, 17(9):1522–1539, 2008.

[74] G. M. Sanchez, V. Vidal, G. Verdu, P. Mayo, and F. Rodenas. Medical image restoration with different types of noise. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4382–4385, August 2012.

[75] M. Sanuki and T. Sasaki. Computing approximate gcds in ill-conditioned cases. In *Proceedings of International Workshop on Symbolic-Numeric Computation*, SNC '07, pages 170–179, NY, USA, 2007. ACM Press.

[76] B. L. Satherley and C. R. Parker. Two-dimensional image reconstruction from zero sheets. *Optics Letters*, 18(23):2053–2055, December 1993.

[77] A. Schnhage. Quasi-GCD computations. *Journal of Complexity*, 1(1):118 – 137, 1985.

[78] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[79] J. L. Starck, E Pantin, and F Murtagh. Deconvolution in astronomy: A review. *Publications of the Astronomical Society of the Pacific*, 114(800): 1051–1069, 2002.

[80] P. Stoica and T. Söderström. Common factor detection and estimation. *Automatica*, 33:985–989, May 1997.

[81] C. Thorpe, F. Li, Z. Li, Z. Yu, D. Saunders, and J. Yu. A coprime blur scheme for data security in video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):3066–3072, December 2013.

[82] I. Trujillo, A. Aguerri, J. Cepa, and C. M. Gutierrez. The effects of seeing on sersic profiles. ii. the moffat PSF. *Monthly Notices of the Royal Astronomical Society*, 328(3):977–985, September 2001.

[83] C. Vural and W. A. Sethares. Blind image deconvolution via dispersion minimization. *Digital Signal Processing*, 16(2):137–148, March 2006.

[84] J. Winkler, M. Hasan, and X. Lao. Two methods for the calculation of the degree of an approximate greatest common divisor of two inexact polynomials. *Calcolo*, pages 1–27, January 2012.

[85] J. R. Winkler. Blind image deconvolution. Technical report, The University of Sheffield, July 2012.

[86] J. R. Winkler and J. D. Allan. Structured total least norm and approximate GCDs of inexact polynomials. *Journal of Computational and Applied Mathematics*, 215:1–13, May 2008.

[87] J. R. Winkler and M. Hasan. A non-linear structure preserving matrix method for the low rank approximation of the Sylvester resultant matrix. *Journal of Computational and Applied Mathematics*, 234(12):3226–3242, October 2010.

[88] J. R. Winkler and X. Lao. The calculation of the degree of an approximate greatest common divisor of two polynomials. *Journal of Computational and Applied Mathematics*, 235(6):1587–1603, January 2011.

[89] J. Wu, C. Chang, and C. Chen. An improved Richardson-Lucy algorithm for single image deblurring using local extrema filtering. In *International Symposium on Intelligent Signal Processing and Communications Systems*, pages 27–32, November 2012.

[90] L. Yang and J. Ren. Remote sensing image restoration using estimated point spread function. In *International Conference on Information Networking and Automation*, volume 1, pages 48–52, October 2010.

[91] N. Yang. *Structured Matrix Methods for Computations on Bernstein Basis Polynomials*. PhD thesis, Department of Computer Science, Sheffield, England, 2013.

[92] C. J. Zarowski, X. Ma, and F. W. Fairman. QR-factorization method for computing the greatest common divisor of polynomials with inexact coefficients. *IEEE Transactions on Signal Processing*, 48(11):3042–3051, November 2000.

[93] Z. Zeng and B. H. Dayton. The approximate GCD of inexact polynomials. In *Proceedings of ISSAC*, ISSAC '04, pages 320–327, NY, USA, 2004. ACM Press.