Proceedings Paper:
Winkler, J.R. (2015) The Sylvester Resultant Matrix and Image Deblurring. In: Boissonnat, J.D., Lyche, T., Cohen, A., Mazure, M.L., Gibaru, O., Schumaker, L.L. and Gout, C., (eds.) Curves and Surfaces. 8th International Conference on Curves and Surfaces, June 12 - 18, 2014, Paris, France. Lecture Notes in Computer Science (9213). Springer, 461 - 490 . ISBN: 978-3-319-22804-4. ISSN: 0302-9743. EISSN: 1611-3349.

https://doi.org/10.1007/978-3-319-22804-4

The Sylvester resultant matrix and image deblurring

Joab R. Winkler

Department of Computer Science, The University of Sheffield,
Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom
{j.r.winkler@sheffield.ac.uk}

Abstract. This paper describes the application of the Sylvester resultant matrix to image deblurring. In particular, an image is represented as a bivariate polynomial and it is shown that operations on polynomials, specifically greatest common divisor (GCD) computations and polynomial divisions, enable the point spread function to be calculated and an image to be deblurred. The GCD computations are performed using the Sylvester resultant matrix, which is a structured matrix, and thus a structure-preserving matrix method is used to obtain a deblurred image. Examples of blurred and deblurred images are presented, and the results are compared with the deblurred images obtained from other methods.

Keywords: Image deblurring, Sylvester resultant matrix

1 Introduction

The removal of blur from an image is one of the most important problems in image processing, and it is motivated by the many applications, which include medicine, astronomy and microscopy, in which its need arises. If the function that represents the blur, called the point spread function (PSF), is known, then a deblurred image can be computed from the PSF and a blurred form of the exact image. If, however, the PSF is known partially or not at all, additional information, for example, prior information on the image to be deblurred, must be specified. Even if the PSF is known, the computation of a deblurred image is ill-conditioned, and thus regularisation must be applied in order to obtain a stable deblurred image.

It is assumed in this paper that the PSF $\mathcal{H}$ is spatially invariant, in which case a blurred image $\mathcal{G}$ is formed by the convolution of $\mathcal{H}$ and the exact image $\mathcal{F}$, and the addition of noise $\mathcal{N}$,

$$\mathcal{G} = \mathcal{H} \otimes \mathcal{F} + \mathcal{N}, \tag{1}$$

where the PSF is assumed to be known exactly, and thus additive noise is the only source of error (apart from roundoff errors due to floating point arithmetic). The exact image $\mathcal{F}$ and PSF $\mathcal{H}$ are of orders $M \times N$ pixels and $(p+1) \times (r+1)$

pixels respectively, and the blurred image is therefore $(M+p)\times(N+r)$ pixels. Equation (1) can be written as a linear algebraic equation,

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \tag{2}$$

where $\mathbf{g}, \mathbf{f}, \mathbf{n} \in \mathbb{R}^m, \mathbf{H} \in \mathbb{R}^{m\times m}$, $m = MN$, and the vectors $\mathbf{g}, \mathbf{f}$ and $\mathbf{n}$ store the blurred image, the exact image and the added noise, respectively, and the entries of $\mathbf{H}$ are functions of the PSF [10]. The matrix $\mathbf{H}$ is ill-conditioned, and thus a simple solution of (2) will have a large error, and an additional difficulty is introduced by the large size of $\mathbf{H}$, even for small values of M and N. This large size would have considerable implications on the execution time of the computation of the solution $\mathbf{f}$ of (2), but it is shown in [10] that the spatial invariance of the PSF implies that $\mathbf{H}$ is a structured matrix, such that only a small part of $\mathbf{H}$ need be stored and computationally efficient algorithms that exploit its structure can be used. Furthermore, it is shown in [10] that regularisation procedures can be included in these structured matrix algorithms, and thus a stable solution of (2) can be computed rapidly.

It follows from (1) that the computation of $\mathcal{F}$ reduces to the deconvolution of the PSF from the blurred image $\mathcal{G}$ in the presence of noise. This computation is an example of linear deconvolution because the PSF is known, and other methods for linear deconvolution include the Wiener filter, which assumes the PSF, and power spectral densities of the noise and true image, are known, and the Lucy-Richardson algorithm. This algorithm arises from the method of maximum likelihood in which the pixel values of the exact image are assumed to have a Poisson distribution, which is appropriate for photon noise in the image [9].

Methods for linear deconvolution are not appropriate when the PSF is not known, in which case the computation of a deblurred image is called blind image deconvolution (BID). It follows from (1) and (2) that BID is substantially more difficult than linear deconvolution because it reduces to the separation of two convolved signals that are either unknown or partially known. Some methods for BID use a statistical approach, based on the method of maximum likelihood, but unlike linear deconvolution, the PSF is also estimated. Other methods for BID include constrained optimisation [4], autoregressive moving average parameter estimation and deterministic image constraints restoration techniques [13], and zero sheet separation [20, 23].

This paper considers the application of the Sylvester resultant matrix to the calculation of the PSF and its deconvolution from a blurred image.[1] This matrix, which is used extensively for polynomial computations, has been used for BID [5, 16, 19], and the Bézout matrix, which is closely related to the Sylvester matrix, has also been used for BID [15]. The work described in this paper differs, however, from the works in these references in several ways. Specifically, the z and Fourier transforms are used to calculate the PSF in [15, 16, 19], but it is shown in this paper that these transforms are not required because the pixel values of the exact, blurred and deblurred images, and the PSF, are the coefficients of polynomials, and polynomial computations, using the Sylvester matrix, are used

[1] This matrix will, for brevity, henceforth be called the Sylvester matrix.

to deblur an image. This matrix is structured, and a structure-preserving matrix method [22] is therefore used in this work to obtain a deblurred image.

The signal-to-noise ratio (SNR) is required in [5, 15] for the calculation of the degrees in x (column) and y (row) of the bivariate polynomial $H(x, y)$ that represents the PSF because this ratio is a threshold in a stopping criterion in an algorithm for deblurring an image. A different approach is used in [16, 19] because a few trial experiments, with PSFs of various sizes, are performed and visual inspection of the deblurred images is used to determine estimates of the horizontal and vertical extents of the PSF. In this paper, the given blurred image is preprocessed and it is shown this allows the degrees in x and y of $H(x, y)$ to be calculated, such that knowledge of the noise level is not required. This is an important feature of the work described in this paper because the noise level, or the SNR, may not be known, or they may only be known approximately. Furthermore, even if the noise level or the SNR are known, it cannot be assumed they are uniformly distributed across the blurred image. This non-uniform distribution implies that the assumptions of a constant noise level or constant SNR may yield poor results when one or both of these constants are used in normwise termination criteria in an algorithm for deblurring an image.

It is shown in Section 2 that the convolution operation defines the formation of a blurred image and the multiplication of two polynomials. This common feature allows polynomial operations, in particular greatest common divisor (GCD) computations and polynomial divisions, to be used for image deblurring. Properties of the PSF that must be considered for polynomial computations are described in Section 3, and it is shown in Section 4 that the PSF is equal to an approximate GCD (AGCD) of two polynomials. This leads to Section 5, in which the computation of an AGCD of two polynomials using their Sylvester matrix is described.

It is assumed the PSF is separable, and the extension of the method discussed in this paper from a separable PSF to a non-separable PSF is considered in Section 6. It is shown it is necessary to perform more computations of the same kind, and additional computations, when a non-separable PSF is used. Section 7 contains examples in which the deblurred images obtained using the method discussed in this paper are compared with the deblurred images obtained using other methods. The paper is summarised in Section 8.

2 The convolution operation

The multiplication of two polynomials reduces to the convolution of their coefficients, and it follows from (1) that, in the absence of noise, a blurred image is formed by the convolution of the exact image and a spatially invariant PSF. This equivalence between bivariate polynomial multiplication and the formation of a blurred image by this class of PSF is quantified by considering these two operations separately, and then showing that they yield the same result. It follows that if the blurred image $\mathcal{G}$ and the PSF $\mathcal{H}$ are represented by bivariate poly-

nomials, then the deblurred image is formed by the division of the polynomial form of $\mathcal{G}$ by the polynomial form of $\mathcal{H}$.

Consider initially bivariate polynomial multiplication. In particular, let the pixel values $f(i,j)$ of $\mathcal{F}$ be the coefficients of a bivariate polynomial $F(x,y)$ that is of degrees $M-1$ and $N-1$ in x and y respectively,

$$F(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) x^{M-1-i} y^{N-1-j},$$

and let the pixel values $h(k,l)$ of the PSF be the coefficients of a bivariate polynomial $H(x,y)$ that is of degrees p and r in x and y respectively,

$$H(x,y) = \sum_{k=0}^{p} \sum_{l=0}^{r} h(k,l) x^{p-k} y^{r-l}. \tag{3}$$

The product of these polynomials is $G_1(x,y) = F(x,y)H(x,y)$,

$$G_1(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{p} \sum_{l=0}^{r} f(i,j) h(k,l) x^{M+p-1-(i+k)} y^{N+r-1-(j+l)},$$

and the substitutions $s = i+k$ and $t = j+l$ yield

$$G_1(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{s=i}^{p+i} \sum_{t=j}^{r+j} f(i,j) h(s-i,t-j) x^{M+p-1-s} y^{N+r-1-t}.$$

It follows that the coefficient of $x^{M+p-1-s} y^{N+r-1-t}$ in $G_1(x,y)$ is

$$g_1(s,t) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) h(s-i,t-j), \tag{4}$$

where $h(k,l) = 0$ if $k < 0$ or $l < 0$.

Consider now the formation of a blurred image by the convolution of $\mathcal{F}$ and $\mathcal{H}$, as shown in (1). A PSF $p(i,s,j,t)$ quantifies the extent to which the pixel value $f(i,j)$ at position (i,j) in $\mathcal{F}$ influences the pixel value $g_2(s,t)$ at position (s,t) in $\mathcal{G}$. The pixel value $g_2(s,t)$ is therefore given by

$$g_2(s,t) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) p(i,s,j,t), \tag{5}$$

and if the PSF depends on the relative positions of the pixels in $\mathcal{F}$ and $\mathcal{G}$, and not on their absolute positions, then the PSF is spatially invariant. It follows that a spatially invariant PSF is a function of $s-i$ and $t-j$,

$$p(i,s,j,t) = h(s-i,t-j), \tag{6}$$

and thus (5) becomes

$$g_2(s,t) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j)h(s-i,t-j). \tag{7}$$

It follows from (4) and (7) that $g_1(s,t) = g_2(s,t)$, and thus the blurring of an image $\mathcal{F}$ by a spatially invariant PSF $\mathcal{H}$ is equivalent to the multiplication of the polynomial forms of $\mathcal{F}$ and $\mathcal{H}$.

The equivalence of polynomial multiplication and the formation of a blurred image $\mathcal{G}$ shows that $\mathcal{G}$ is larger than the exact image $\mathcal{F}$. Specifically, $\mathcal{F}$ is represented by a polynomial of degrees $M-1$ in x (column) and $N-1$ in y (row), and the PSF is represented by a polynomial of degrees p in x and r in y. It therefore follows that $\mathcal{G}$ is represented by a polynomial of degrees $M+p-1$ and $N+r-1$ in x and y respectively, and thus $\mathcal{G}$ has p and r more pixels than $\mathcal{F}$ along the columns and rows, respectively. These extra pixels are removed after the PSF is computed, when it is deconvolved from $\mathcal{G}$, thereby yielding a deblurred image that is the same size as $\mathcal{F}$.

The polynomial computations that allow a PSF to be calculated require that the different forms of a PSF be considered because they affect these computations. This issue is addressed in the next section.

3 The point spread function

The applicability of polynomial computations to blind image deconvolution assumes the PSF is spatially invariant (6), as shown by the equivalence of (4) and (7). A spatially invariant PSF may be separable or non-separable, and it follows from (3) that a PSF is separable if

$$H(x,y) = H_c(x)H_r(y), \tag{8}$$

where $H_c(x)$ and $H_r(y)$ are the column and row blurring functions respectively. The assumption of separability is more restrictive than the assumption of spatial invariance, but it is included in the work described in this paper because, as discussed in Section 6, the removal of the separability condition introduces additional computations.

Spatial invariance of the PSF is satisfied in many problems, but there also exist problems in which it is not satisfied, and thus a spatially variant PSF must sometimes be considered. The mathematical implications of a spatially variant PSF are considerable because (5) does not reduce to (7) in this circumstance since (6) is not satisfied by a spatially variant PSF, and thus the convolution operation is not appropriate. Furthermore, the matrix $\mathbf{H}$ in (2) is not structured if the PSF is spatially variant, which has obvious implications for the computational cost, with respect to complexity and memory requirements, of the algorithm used for the solution of (2). It is therefore desirable to retain in a deblurring algorithm that uses a spatially variant PSF the computational properties associated with

a spatially invariant PSF. Nagy *et. al.* [17] address this issue by considering a class of spatially variant PSFs that are formed by the addition of several spatially invariant PSFs, where each spatially invariant PSF is restricted to a small subregion of the blurred image. Piecewise constant interpolation is used to join these spatially invariant PSFs in each subregion in order to form a spatially variant PSF.

4 Polynomial computations for image deblurring

It was shown in Section 2 that the formation of a blurred image by a spatially invariant PSF can be represented by the multiplication of the bivariate polynomials that represent the exact image and the PSF. This polynomial multiplication is considered in detail in this section and it is shown that the PSF is equal to an AGCD of two polynomials.

It follows from Section 2 that (1) can be written as

$$G(x,y) = H(x,y)F(x,y) + N(x,y), \tag{9}$$

where the PSF satisfies (8), and thus if it is assumed $E(x,y)$ is the uncertainty in the PSF, then (9) is generalised to

$$G(x,y) = \left(H_c(x)H_r(y) + E(x,y)\right)F(x,y) + N(x,y). \tag{10}$$

Consider two rows $x = r_1$ and $x = r_2$, and two columns $y = c_1$ and $y = c_2$, of $\mathcal{G}$,

$$\begin{aligned}
G(r_1,y) &= \left(H_c(r_1)H_r(y) + E(r_1,y)\right)F(r_1,y) + N(r_1,y),\\
G(r_2,y) &= \left(H_c(r_2)H_r(y) + E(r_2,y)\right)F(r_2,y) + N(r_2,y),\\
G(x,c_1) &= \left(H_c(x)H_r(c_1) + E(x,c_1)\right)F(x,c_1) + N(x,c_1),\\
G(x,c_2) &= \left(H_c(x)H_r(c_2) + E(x,c_2)\right)F(x,c_2) + N(x,c_2),
\end{aligned}$$

where $H_c(r_1), H_c(r_2), H_r(c_1)$ and $H_r(c_2)$ are constants. It is adequate to consider either the equations for the rows r_1 and r_2, or the equations for the columns c_1 and c_2, because they have the same form. Consider, therefore, the equations for r_1 and r_2, which are equations in the independent variable y,

$$\begin{aligned}
G(r_1,y) &= \left(H_c(r_1)H_r(y) + E(r_1,y)\right)F(r_1,y) + N(r_1,y),\\
G(r_2,y) &= \left(H_c(r_2)H_r(y) + E(r_2,y)\right)F(r_2,y) + N(r_2,y).
\end{aligned} \tag{11}$$

The polynomials $E(r_1,y)$ and $E(r_2,y)$ represent uncertainties in the PSF, and if their magnitudes are sufficiently small, then (11) can be written as

$$\begin{aligned}
G(r_1,y) &\approx H_c(r_1)H_r(y)F(r_1,y) + N(r_1,y),\\
G(r_2,y) &\approx H_c(r_2)H_r(y)F(r_2,y) + N(r_2,y).
\end{aligned} \tag{12}$$

Also, if the magnitudes of the polynomials $N(r_1,y)$ and $N(r_2,y)$ that represent the noise are sufficiently small, then the approximations (12) simplify to

$$\begin{aligned}
G(r_1,y) &\approx H_c(r_1)H_r(y)F(r_1,y),\\
G(r_2,y) &\approx H_c(r_2)H_r(y)F(r_2,y).
\end{aligned}$$

It follows that if the polynomials $F(r_1, y)$ and $F(r_2, y)$, that is, the polynomial forms of the rows r_1 and r_2 of $\mathcal{F}$, are coprime, then

$$H_r(y) = \text{AGCD}\,(G(r_1, y), G(r_2, y)). \tag{13}$$

The arguments r_1 and r_2 appear on the right hand side of this equation, and not on the left hand side, because it is assumed the PSF is separable, and thus r_1 and r_2 are the indices of any two rows of $\mathcal{G}$.

Equation (13) shows that the row component of a separable and spatially invariant PSF is equal to an AGCD of any two rows of a blurred image if the pixel values of these rows are the coefficients of two polynomials. It is clear that the column component $H_c(x)$ of a separable and spatially invariant PSF can be computed identically, and the PSF can then be calculated from (8). After the PSF components $H_c(x)$ and $H_r(y)$ have been calculated, the deblurred image $\tilde{\mathcal{F}}$ is calculated from an approximate form of (10),

$$\tilde{F}(x, y) \approx \frac{G(x, y)}{H_c(x) H_r(y)}, \tag{14}$$

which involves two approximate polynomial divisions.

Consider initially the approximate division for the calculation of $Q(x, y)$, which is the polynomial representation of the partially deblurred image $\mathcal{Q}$ formed after the row component of the PSF has been deconvolved from $\mathcal{G}$,

$$Q(x, y) \approx \frac{G(x, y)}{H_r(y)}. \tag{15}$$

It follows that $H_r(y)Q(x, y) \approx G(x, y)$, and this approximate polynomial equation is applied to all the rows of $\mathcal{G}$. This approximate equation is written in matrix form,

$$\mathbf{T}_r \mathbf{q}_i \approx \mathbf{g}_i, \qquad i = 1, \ldots, M + p, \tag{16}$$

where $\mathbf{T}_r \in \mathbb{R}^{(N+r)\times N}$ is a lower triangular Tœplitz matrix whose entries on and below the leading diagonal are the coefficients of $H_r(y)$, $\mathbf{q}_i \in \mathbb{R}^N$ is the transpose of the ith row of the partially deblurred image $\mathcal{Q}$, $\mathbf{g}_i \in \mathbb{R}^{N+r}$ is the transpose of the ith row of $\mathcal{G}$, and p and r are defined in (3). The approximations (16) are solved, in the least squares sense, for each vector $\mathbf{q}_i$ and thus the partially deblurred image $\mathcal{Q}$, which is of order $(M + p) \times N$ pixels, is obtained.

The column component of the PSF must be deconvolved from $\mathcal{Q}$ in order to obtain the deblurred image $\tilde{\mathcal{F}}$, and it follows from (14) and (15) that $\tilde{\mathcal{F}}$ is computed from the approximation

$$H_c(x)\tilde{F}(x, y) \approx Q(x, y),$$

which is applied to all the columns of $\mathcal{Q}$. These N approximations can be written in a form that is identical to (16),

$$\mathbf{T}_c \tilde{\mathbf{f}}_j \approx \mathbf{q}_j, \qquad j = 1, \ldots, N, \tag{17}$$

where $\mathbf{T}_c \in \mathbb{R}^{(M+p)\times M}$ is a Tœplitz matrix that is similar to $\mathbf{T}_r$ except that its entries are the coefficients of the column component $H_c(x)$ of the PSF, $\tilde{\mathbf{f}}_j \in \mathbb{R}^M$ is the jth column of $\tilde{\mathcal{F}}$ and $\mathbf{q}_j \in \mathbb{R}^{M+p}$ is the jth column of the partially deblurred image $\mathcal{Q}$. It follows that the image $\tilde{\mathcal{F}}$ that results from the approximate polynomial divisions (14) is formed from the vectors $\tilde{\mathbf{f}}_j, j = 1, \ldots, N$,

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{\mathbf{f}}_1 & \tilde{\mathbf{f}}_2 & \cdots & \tilde{\mathbf{f}}_{N-1} & \tilde{\mathbf{f}}_N \end{bmatrix} \in \mathbb{R}^{M\times N}.$$

The analysis in this section shows that the computation of a separable and spatially invariant PSF involves two AGCD computations. The computation of an AGCD of two polynomials must be done with care because it is an ill-posed operation, and it is considered in the next section.

5 Approximate greatest common divisors

This section considers the computation of an AGCD of two polynomials from their Sylvester matrix and subresultant matrices. This computation is required for the determination of the column and row components of the PSF, as shown in (13) for the row component. It is appropriate, however, to consider initially exact polynomials that have a non-constant GCD, and then describe the modifications required when inexact forms of these polynomials, which have an AGCD, are considered.

Let $\hat{p}(y)$ and $\hat{q}(y)$ be exact polynomials of degrees r and s respectively,[2] and let $\hat{d}(y) = \mathrm{GCD}\,(\hat{p}, \hat{q})$, where $\hat{t} = \deg \hat{d}(y) > 0$,

$$\hat{p}(y) = \sum_{i=0}^{r} \hat{p}_i y^{r-i} \qquad \text{and} \qquad \hat{q}(y) = \sum_{i=0}^{s} \hat{q}_i y^{s-i}. \tag{18}$$

It is shown in [1] that $\hat{d}(y)$ can be computed from the Sylvester matrix $\mathbf{S}(\hat{p}, \hat{q})$ of $\hat{p}(y)$ and $\hat{q}(y)$, and its subresultant matrices $\mathbf{S}_k(\hat{p}, \hat{q}), k = 2, \ldots, \min(r, s)$, where $\mathbf{S}_1(\hat{p}, \hat{q}) = \mathbf{S}(\hat{p}, \hat{q})$ and $\mathbf{S}_k(\hat{p}, \hat{q}) \in \mathbb{R}^{(r+s-k+1)\times(r+s-2k+2)}, k = 1, \ldots, \min(r, s)$. The Sylvester matrix $\mathbf{S}(\hat{p}, \hat{q})$ is

$$\mathbf{S}(\hat{p}, \hat{q}) = \begin{bmatrix} \hat{p}_0 & & & & \hat{q}_0 & & & \\ \hat{p}_1 & \hat{p}_0 & & & \hat{q}_1 & \hat{q}_0 & & \\ \vdots & \hat{p}_1 & \ddots & & \vdots & \hat{q}_1 & \ddots & \\ \hat{p}_{r-1} & \vdots & \ddots & \hat{p}_0 & \hat{q}_{s-1} & \vdots & \ddots & \hat{q}_0 \\ \hat{p}_r & \hat{p}_{r-1} & \ddots & \hat{p}_1 & \hat{q}_s & \hat{q}_{s-1} & \ddots & \hat{q}_1 \\ & \hat{p}_r & \ddots & \vdots & & \hat{q}_s & \ddots & \vdots \\ & & \ddots & \hat{p}_{r-1} & & & \ddots & \hat{q}_{s-1} \\ & & & \hat{p}_r & & & & \hat{q}_s \end{bmatrix},$$

[2] The degree r of $\hat{p}(y)$ is not related to the degree r in y of $H(x, y)$, which is defined in (3).

where the coefficients of $\hat{p}(y)$ and $\hat{q}(y)$ occupy the first s columns and last r columns, respectively. It is seen that $\mathbf{S}(\hat{p}, \hat{q})$ has a partitioned structure, and it is shown in the sequel that this property may cause numerical problems. The subresultant matrices $\mathbf{S}_k(\hat{p}, \hat{q}), k = 2, \ldots, \min(r, s)$, are formed by deleting rows and columns from $\mathbf{S}(\hat{p}, \hat{q})$, and they also have a partitioned structure that must be considered when computational issues are addressed. The application of these matrices to the calculation of the GCD of $\hat{p}(y)$ and $\hat{q}(y)$ is based on the following theorem.

Theorem 1. *Let $\hat{p}(y)$ and $\hat{q}(y)$ be defined in (18).*

1. *The degree $\hat{t}$ of the GCD of $\hat{p}(y)$ and $\hat{q}(y)$ is equal to the rank loss of $\boldsymbol{S}(\hat{p}, \hat{q})$,*
$$\hat{t} = r + s - \mathit{rank}\, \boldsymbol{S}(\hat{p}, \hat{q}).$$
2. *The coefficients of $\hat{d}(y)$ are contained in the last non-zero rows of the upper triangular matrices $\boldsymbol{U}$ and $\boldsymbol{R}$ obtained from, respectively, the LU and QR decompositions of $\boldsymbol{S}(\hat{p}, \hat{q})^T$.*
3. *The value of $\hat{t}$ is equal to the largest integer k such that the kth subresultant matrix $\boldsymbol{S}_k(\hat{p}, \hat{q})$ is singular,*
$$\begin{aligned} &\mathit{rank}\, \boldsymbol{S}_k(\hat{p}, \hat{q}) < r + s - 2k + 2, \;\; k = 1, \ldots, \hat{t}, \\ &\mathit{rank}\, \boldsymbol{S}_k(\hat{p}, \hat{q}) = r + s - 2k + 2, \;\; k = \hat{t} + 1, \ldots, \min(r, s), \end{aligned}$$
where
$$\boldsymbol{S}_k(\hat{p}, \hat{q}) = \left[\boldsymbol{C}_k(\hat{p}) \; \boldsymbol{D}_k(\hat{q}) \right], \tag{19}$$
and $\boldsymbol{C}_k(\hat{p}) \in \mathbb{R}^{(r+s-k+1)\times(s-k+1)}$ and $\boldsymbol{D}_k(\hat{q}) \in \mathbb{R}^{(r+s-k+1)\times(r-k+1)}$ are Toeplitz matrices.

It follows from the definition of $\hat{d}(y)$ that there exist coprime polynomials $\hat{u}(y)$ and $\hat{v}(y)$, of degrees $r - \hat{t}$ and $s - \hat{t}$ respectively, that satisfy
$$\hat{p}(y) = \hat{u}(y)\hat{d}(y) \qquad \text{and} \qquad \hat{q}(y) = \hat{v}(y)\hat{d}(y). \tag{20}$$
The next lemma considers the rank of the $\hat{t}$th subresultant matrix $\mathbf{S}_{\hat{t}}(\hat{p}, \hat{q})$.

Lemma 1. *The rank of $\boldsymbol{S}_{\hat{t}}(\hat{p}, \hat{q})$ is equal to $r + s - 2\hat{t} + 1$.*

Proof. Since the degree of the GCD of $\hat{p}(y)$ and $\hat{q}(y)$ is $\hat{t}$, there exists exactly one set of polynomials $\hat{u}(y)$ and $\hat{v}(y)$, defined up to an arbitrary non-zero scalar multiplier, that satisfies (20). The elimination of $\hat{d}(y)$ between the equations in (20) yields an equation, the matrix form of which is
$$\mathbf{S}_{\hat{t}}(\hat{p}, \hat{q}) \begin{bmatrix} \hat{\mathbf{v}} \\ -\hat{\mathbf{u}} \end{bmatrix} = \mathbf{0},$$
where $\mathbf{S}_k(\hat{p}, \hat{q})$ is defined in (19), and $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ are, respectively, vectors of the coefficients of $\hat{u}(y)$ and $\hat{v}(y)$. Since $\hat{u}(y)$ and $\hat{v}(y)$ are unique up to an arbitrary non-zero scalar multiplier, the dimension of the null space of $\mathbf{S}_{\hat{t}}(\hat{p}, \hat{q})$ is one, and it therefore follows that the rank of $\mathbf{S}_{\hat{t}}(\hat{p}, \hat{q})$ is $r + s - 2\hat{t} + 1$.

The exact polynomials $\hat{p}(y)$ and $\hat{q}(y)$ are subject to errors in practical problems and thus inexact forms of these polynomials, $p(y)$ and $q(y)$ respectively, must be considered. These polynomials are, with high probability, coprime, but an algorithm that returns $\deg \mathrm{GCD}\,(p, q) = 0$ is unsatisfactory because this result is governed entirely by the errors in the coefficients of $p(y)$ and $g(y)$, and it does not consider the proximity of $(p(y), q(y))$ to $(\hat{p}(y), \hat{q}(y))$. An effective algorithm for the computation of an AGCD of $p(y)$ and $q(y)$ should return

$$\mathrm{AGCD}\,(p, q) \approx \mathrm{GCD}\,(\hat{p}, \hat{q}),$$

such that the error in this approximation is small.

The computation of an AGCD is performed in two stages:

Stage 1: Compute the degree t of an AGCD of $p(y)$ and $q(y)$.
Stage 2: Compute the coefficients of an AGCD of degree t.

It is shown in [28] that $p(y)$ and $q(y)$ must be processed by three operations before their Sylvester matrix $\mathbf{S}(p, q)$ is used to compute an AGCD. The first preprocessing operation arises because $\mathbf{S}(p, q)$ has, as noted above, a partitioned structure, which may cause numerical problems if the coefficients of $p(y)$ are much smaller or larger in magnitude than the coefficients of $q(y)$ since $\mathbf{S}(p, q)$ is not balanced if this condition is satisfied. It is therefore necessary to normalise $p(y)$ and $q(y)$, and Theorem 2 shows that the normalisation of an arbitrary polynomial $s(y)$,

$$s(y) = \sum_{i=0}^{m} s_i y^{m-i}, \tag{21}$$

by the geometric mean of its coefficients is better than the normalisation by the 2-norm of the vector $\mathbf{s}$ of its coefficients.

Theorem 2. *Consider the polynomial $s(y)$, which is defined in (21), and the polynomials $s_1(y)$ and $s_2(y)$ that are formed from $s(y)$ by two different normalisations,*

$$s_1(y) = \frac{s(y)}{\|\boldsymbol{s}\|_2} \qquad \textit{and} \qquad s_2(y) = \frac{s(y)}{(\prod_{i=0}^{m} s_i)^{\frac{1}{m+1}}},$$

where it is assumed, for simplicity, $s_i > 0$. The ith coefficients of $s_1(y)$ and $s_2(y)$ are therefore

$$s_{1,i} = \frac{s_i}{\|\boldsymbol{s}\|_2} \qquad \textit{and} \qquad s_{2,i} = \frac{s_i}{(\prod_{i=0}^{m} s_i)^{\frac{1}{m+1}}},$$

and the relative errors of the ith coefficients of $s(y), s_1(y)$ and $s_2(y)$ when the coefficients of $s(y)$ are perturbed are, respectively,

$$\Delta s_i = \frac{|\delta s_i|}{s_i}, \qquad \Delta s_{1,i} = \frac{|\delta s_{1,i}|}{s_{1,i}} \qquad \textit{and} \qquad \Delta s_{2,i} = \frac{|\delta s_{2,i}|}{s_{2,i}}.$$

The ratio of the relative error of $s_{1,i}$ to the relative error of s_i is

$$r_1(s_i) = \frac{\Delta s_{1,i}}{\Delta s_i} = 1 - \frac{s_i^2}{\|\boldsymbol{s}\|_2^2},$$

and the ratio of the relative error of $s_{2,i}$ to the relative error of s_i is

$$r_2 = \frac{\Delta s_{2,i}}{\Delta s_i} = \frac{m}{m+1}.$$

It follows that $r_1(s_i)$ is a function of the coefficients s_i, and thus this form of normalisation may change the relative errors of the coefficients on which computations are performed. This is different from r_2, which is constant and independent of these coefficients, and thus normalisation by the geometric mean of the coefficients of $s(y)$ is preferred. This form of normalisation is therefore used in the work described in this paper.

It follows that the normalised forms of $p(y)$ and $q(y)$, which are inexact forms of the exact polynomials $\hat{p}(y)$ and $\hat{q}(y)$ that are defined in (18), are

$$\dot{p}(y) = \sum_{i=0}^{r} \bar{p}_i y^{r-i}, \qquad \bar{p}_i = \frac{p_i}{(\prod_{i=0}^{r} |p_i|)^{\frac{1}{r+1}}}, \qquad p_i \neq 0, \tag{22}$$

and

$$\dot{q}(y) = \sum_{i=0}^{s} \bar{q}_i y^{s-i}, \qquad \bar{q}_i = \frac{q_i}{(\prod_{i=0}^{s} |q_i|)^{\frac{1}{s+1}}}, \qquad q_i \neq 0, \tag{23}$$

and it is clear that this computation must be changed if the non-zero condition on the coefficients of $p(y)$ or $q(y)$ is not satisfied. It was noted above that this normalisation defines the first of three operations that must be implemented before computations are performed on $\mathbf{S}(p, q)$, and the second and third preprocessing operations are now considered.

The second preprocessing operation arises because an AGCD of $\dot{p}(y)$ and $\dot{q}(y)$ is a function of their coefficients, and it is independent of the magnitudes of their coefficient vectors. It therefore follows that if α is an arbitrary non-zero constant, then

$$\text{AGCD}(\dot{p}, \dot{q}) \sim \text{AGCD}(\dot{p}, \alpha\dot{q}), \tag{24}$$

where $\sim$ denotes equivalence to within an arbitrary non-zero scalar multiplier. The optimal value of the constant α must be calculated, and this issue is addressed below.

It is shown in [6] that the ratio of the entry of maximum magnitude of an arbitrary matrix X to the entry of minimum magnitude of X is believed to be a useful condition number of X, and it is therefore desirable to minimise this ratio. The objective of the third preprocessing operation is therefore the minimisation

of the ratio $\mathcal{R}$ of the entry of maximum magnitude, to the entry of minimum magnitude, of the Sylvester matrix. A change in the independent variable from y to w is made,

$$y = \theta w, \tag{25}$$

and α and θ are constants whose optimal values minimise $\mathcal{R}$. It follows from (22), (23), (24) and (25) that an AGCD of the polynomials

$$\bar{p}(w,\theta) = \sum_{i=0}^{r} \left(\bar{p}_i \theta^{r-i}\right) w^{r-i} \qquad \text{and} \qquad \alpha\bar{q}(w,\theta) = \alpha \sum_{i=0}^{s} \left(\bar{q}_i \theta^{s-i}\right) w^{s-i},$$

is computed, and thus the optimal values α_0 and θ_0 of α and θ, respectively, minimise $\mathcal{R}$,

$$\alpha_0, \theta_0 = \arg\min_{\alpha,\theta} \left\{ \frac{\max\left\{\max_{i=0,\ldots,r} \left|\bar{p}_i \theta^{r-i}\right|, \max_{j=0,\ldots,s} \left|\alpha\bar{q}_j \theta^{s-j}\right|\right\}}{\min\left\{\min_{i=0,\ldots,r} \left|\bar{p}_i \theta^{r-i}\right|, \min_{j=0,\ldots,s} \left|\alpha\bar{q}_j \theta^{s-j}\right|\right\}} \right\}.$$

It is shown in [26] that this minimisation problem can be transformed to a linear programming (LP) problem from which α_0 and θ_0 are easily computed. The arguments of this problem are the coefficients $\bar{p}_i$ and $\bar{q}_j$ of, respectively, the noisy polynomials $\bar{p}(w,\theta)$ and $\bar{q}(w,\theta)$, and it is therefore necessary that α_0 and θ_0 be insensitive to noise. The sensitivity of the LP problem to perturbations in its arguments is addressed in [18], and some guidelines for performing sensitivity analysis are stated.

The LP problem is solved by an iterative procedure, and its stability must therefore be considered. In particular, if the iterations fail to converge or the maximum number of iterations is exceeded, then the values $\alpha_0 = 1$ and $\theta_0 = 1$ are used, that is, the only preprocessing operation is the normalisation of the coefficients of $p(y)$ and $q(y)$. Numerous computational experiments on blurred images showed, however, that the LP problem always converged, even with images of about 500×500 pixels and PSFs whose degrees in x and y are large.

It follows that an AGCD of the given inexact polynomials $p(y)$ and $q(y)$ is computed from the Sylvester matrix $\mathbf{S}(\bar{p}, \alpha_0\bar{q}) = \mathbf{S}(\bar{p}(w,\theta_0), \alpha_0\bar{q}(w,\theta_0))$. Computational experiments in [28] show the importance of the inclusion of α_0 and θ_0 for AGCD computations, and it is also shown that incorrect values of α_0 or θ_0 may lead to an incorrect result.

The next section considers the computation of the degree and coefficients of an AGCD of $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$ from their Sylvester matrix.

5.1 The Sylvester resultant matrix

This section describes the use of the Sylvester matrix for the computation of an AGCD of $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$. The simplest method of performing this calculation is by using the LU or QR decompositions of $\mathbf{S}(\bar{p}, \alpha_0\bar{q})^T$, as stated in Theorem 1, but these decompositions yield poor results, and a structure-preserving matrix method [22] yields much better results.

Consider Stage 1 of the calculation of an AGCD of $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$, which is, as noted in Section 5, the determination of its degree t. Two methods for the calculation of t that use the singular value decomposition (SVD) of the Sylvester matrix of $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$, and its subresultant matrices, are described in [28], and it is shown in more recent work [25] that t can also be computed from the QR decomposition of these matrices. Since the subresultant matrix $\mathbf{S}_{k+1}(\bar{p},\alpha_0\bar{q})$ is formed by the deletion of two columns and one row from the subresultant matrix $\mathbf{S}_k(\bar{p},\alpha_0\bar{q})$, it follows that the update formula of the QR decomposition allows efficient computation, and it is therefore preferred to the SVD, whose update formula is complicated, for the calculation of t.

It follows from Lemma 1 that, using the calculated value of t, the numerical rank loss of $\mathbf{S}_t(\bar{p},\alpha_0\bar{q})$ is one, and there therefore exists a vector $\tilde{\mathbf{x}}$ such that

$$\mathbf{S}_t(\bar{p},\alpha_0\bar{q})\tilde{\mathbf{x}} \approx \mathbf{0}, \qquad \frac{\|\mathbf{S}_t(\bar{p},\alpha_0\bar{q})\tilde{\mathbf{x}}\|}{\|\mathbf{S}_t(\bar{p},\alpha_0\bar{q})\|\,\|\tilde{\mathbf{x}}\|} \ll 1, \tag{26}$$

where the ith column of $\mathbf{S}_t(\bar{p},\alpha_0\bar{q})$ is $\mathbf{c}_i \in \mathbb{R}^{r+s-t+1}$,

$$\mathbf{S}_t(\bar{p},\alpha_0\bar{q}) = \begin{bmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_{q-1} & \mathbf{c}_q & \mathbf{c}_{q+1} & \dots & \mathbf{c}_{r+s-2t+1} & \mathbf{c}_{r+s-2t+2}\end{bmatrix}.$$

It follows from (26) that one column $\mathbf{c}_q$ of $\mathbf{S}_t(\bar{p},\alpha_0\bar{q})$ is almost linearly dependent on the other columns, and the methods for the calculation of t described in [28] also return the index q of the column $\mathbf{c}_q$. It follows that the matrix $\mathbf{A}_q$ formed by the removal of $\mathbf{c}_q$ from $\mathbf{S}_t(\bar{p},\alpha_0\bar{q})$,

$$\mathbf{A}_q = \begin{bmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_{q-1} & \mathbf{c}_{q+1} & \dots & \mathbf{c}_{r+s-2t+1} & \mathbf{c}_{r+s-2t+2}\end{bmatrix} \in \mathbb{R}^{(r+s-t+1)\times(r+s-2t+1)},$$

has full rank, and thus (26) can be written as

$$\mathbf{A}_q\mathbf{x} \approx \mathbf{c}_q, \tag{27}$$

from which it follows that the qth entry of $\tilde{\mathbf{x}}$, which is defined in (26), is equal to -1. It is shown in [26, 27] that the entries of $\mathbf{x}$ are the coefficients of the coprime polynomials $\bar{u}(w,\theta_0)$ and $\bar{v}(w,\theta_0)$, which are of degrees $r-t$ and $s-t$ respectively, and that they satisfy

$$\bar{p}(w,\theta_0) \approx \bar{d}(w,\theta_0)\bar{u}(w,\theta_0) \qquad \text{and} \qquad \alpha_0\bar{q}(w,\theta_0) \approx \bar{d}(w,\theta_0)\bar{v}(w,\theta_0),$$

where $\bar{d}(w,\theta_0)$ is an AGCD of $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$.

It follows from Lemma 1 that $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$ have a GCD of degree t if (26) is cast into an exact equation, that is, the coefficient matrix of a modified form of this equation has unit rank loss exactly and not approximately. This modification is achieved by adding the Sylvester subresultant matrix,

$$\mathbf{T}_t = \mathbf{T}_t(\ddot{p}(w,\theta_0), \alpha_0\ddot{q}(w,\theta_0)),$$

of the polynomials $\ddot{p}(w,\theta_0)$ and $\alpha_0\ddot{q}(w,\theta_0)$, which are defined as

$$\ddot{p}(w,\theta_0) = \sum_{i=0}^{r} \left(\ddot{p}_i\theta_0^{r-i}\right) w^{r-i} \qquad \text{and} \qquad \alpha_0\ddot{q}(w,\theta_0) = \alpha_0 \sum_{i=0}^{s} \left(\ddot{q}_i\theta_0^{s-i}\right) w^{s-i},$$

to $\mathbf{S}_t(\bar{p}, \alpha_0\bar{q})$, such that (26) becomes

$$(\mathbf{S}_t + \mathbf{T}_t)\,\bar{\mathbf{x}} = \mathbf{0}, \tag{28}$$

where $\mathbf{S}_t + \mathbf{T}_t$ is the tth Sylvester subresultant matrix of $\bar{p}(w,\theta_0) + \ddot{p}(w,\theta_0)$ and $\alpha_0\,(\bar{q}(w,\theta_0) + \ddot{q}(w,\theta_0))$, and the rank of $\mathbf{S}_t + \mathbf{T}_t$ is $r + s - 2t + 1$.

The homogeneous equation (28) is transformed to a linear algebraic equation by the removal of the qth column of the coefficient matrix, where the index q is defined in (27), to the right hand side. In particular, if $\mathbf{e}_q$ is the qth column of $\mathbf{T}_t$, and $\mathbf{E}_q$ is formed from the remaining $r + s - 2t + 1$ columns of $\mathbf{T}_t$, then it follows from (27) that (28) becomes

$$(\mathbf{A}_q + \mathbf{E}_q)\,\bar{\mathbf{x}} = \mathbf{c}_q + \mathbf{e}_q, \tag{29}$$

where $\mathbf{A}_q$ and $\mathbf{E}_q$ have the same structure, and $\mathbf{c}_q$ and $\mathbf{e}_q$ have the same structure. The matrix $\mathbf{T}_t$ is not unique because there exists more than one set of polynomials $(\ddot{p}(w,\theta_0), \alpha_0\ddot{q}(w,\theta_0))$ such that $\bar{p}(w,\theta_0)+\ddot{p}(w,\theta_0)$ and $\alpha_0\,(\bar{q}(w,\theta_0) + \ddot{q}(w,\theta_0))$ have a GCD of degree t. Uniqueness is imposed by requiring that, of all the polynomials $\ddot{p}(w,\theta_0)$ and $\alpha_0\ddot{q}(w,\theta_0)$ that can be added to, respectively, $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$, the polynomials $\ddot{p}(w,\theta_0)$ and $\alpha_0\ddot{q}(w,\theta_0)$ of minimum magnitude be sought, that is, the perturbed polynomials must be as near as possible to the transformed forms $\bar{p}(w,\theta_0)$ and $\alpha_0\bar{q}(w,\theta_0)$ of the given inexact polynomials. It is shown in [26] that the imposition of this constraint on (29) yields a non-linear equation, which is solved iteratively. The first order approximation of this equation generates a least squares minimisation subject to an equality constraint, the LSE problem, at each iteration $j = 1, 2, \ldots,$

$$\min_{\delta\mathbf{y}^{(j)}} \left\| \delta\mathbf{y}^{(j)} - \mathbf{h}^{(j)} \right\|_2 \qquad \text{subject to} \qquad \mathbf{P}^{(j)}\delta\mathbf{y}^{(j)} = \mathbf{r}^{(j)}, \tag{30}$$

where

$$\begin{aligned}
\mathbf{y}^{(j)} &= \mathbf{y}^{(j-1)} + \delta\mathbf{y}^{(j)}, && (31)\\
\mathbf{h}^{(j)} &= \mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}, && (32)\\
\mathbf{y}^{(j)} &= \mathbf{y}\Big(\ddot{p}^{(j)}, \ddot{q}^{(j)}, \alpha_0^{(j)}, \theta_0^{(j)}, \bar{\mathbf{x}}^{(j)}\Big),\\
\mathbf{P}^{(j)} &= \mathbf{P}\Big(\bar{p}, \bar{q}, \ddot{p}^{(j)}, \ddot{q}^{(j)}, \alpha_0^{(j)}, \theta_0^{(j)}, \bar{\mathbf{x}}^{(j)}\Big),\\
\mathbf{r}^{(j)} &= \mathbf{r}\Big(\bar{p}, \bar{q}, \ddot{p}^{(j)}, \ddot{q}^{(j)}, \alpha_0^{(j)}, \theta_0^{(j)}, \bar{\mathbf{x}}^{(j)}\Big),
\end{aligned}$$

$\alpha_0^{(0)} = \alpha_0$ and $\theta_0^{(0)} = \theta_0$. The LSE problem (30) is solved by the QR decomposition and its convergence is considered in Section 5.2.

The blurred image is of order $(M + p) \times (N + r)$ pixels, and it is therefore represented by a polynomial of degrees $M_1 = M + p - 1$ and $N_1 = N + r - 1$ in x and y respectively. The computation of the degree of the row component of the polynomial form of the PSF is therefore calculated from a Sylvester matrix of order $2N_1 \times 2N_1$, and thus $8O(N_1^3)$ flops are required for its QR decomposition.

The computation of the QR decomposition of each subresultant matrix requires $O(N_1^2)$ flops, and since $N_1 - 1$ subresultant matrices are required for the computation of the degree of the row component of the polynomial form of the PSF, it follows that this computation requires a total of $(N_1 - 1)O(N_1^2) = O(N_1^3)$ flops. The repetition of this computation for the column component of the polynomial form of the PSF shows that $9O(N_1^3) + 9O(M_1^3)$ flops are required for the computation of the degrees of the row and column components of the polynomial form of the PSF. Each iteration for the solution of the LSE problem (30) by the QR decomposition requires approximately $(M_1 + N_1)^3$ flops, and numerous computational experiments [27] showed that, even for high levels of noise, convergence of this iterative procedure is achieved in a few iterations. It therefore follows that the AGCD computations required for the solution of the BID problem are cubic in complexity.

It was shown in Section 4 that this AGCD computation is applied to the rows r_1 and r_2 of the blurred image $\mathcal{G}$, thereby yielding a partially deblurred image $\mathcal{Q}$, and it is then repeated for the columns c_1 and c_2 of $\mathcal{Q}$. These computations enable the PSF to be computed, and the deblurred image is then obtained by two deconvolutions, as shown in (16) and (17). The coefficient matrices in these approximate equations are Tœplitz and thus a structure-preserving matrix method [21] can be used to obtain an improved deblurred image. In this work, however, the least squares solutions of (16) and (17) are used because the examples in Section 7 show that these simple solutions yield deblurred images of high quality, and that they are better than the deblurred images obtained from four other methods of image deblurring.

There is an extensive literature on AGCD computations and many methods have been developed, including methods based on the QR decomposition [3, 24, 29, 30] and optimisation [2, 12]. The AGCD in (13) and its column equivalent are computed from a structured low rank approximation of $\mathbf{S}(p, q)$ for the work described in this paper because this approximation exploits the structure of $\mathbf{S}(p, q)$ [26, 27]. A structure-preserving matrix method is also used for the AGCD computation in [11, 14], and it is similar to the method used in this paper. There are, however, three important differences between the method used in [11, 14] and the method used in this paper:

1. The preprocessing operations discussed in Section 5 form an important part of the method for the AGCD computations used in this paper, but preprocessing operations are not used in [11, 14]. Two of these processing operations introduce the parameters α_0 and θ_0, and their inclusion in the problem formulation implies that (29) is a non-linear equation. A non-linear structure-preserving matrix method [22] is therefore used, and it must be compared with the linear structure-preserving matrix method [21] used in [11, 14] because the parameters α_0 and θ_0 are not included in the AGCD computation in these references, which is equivalent to the specification $\alpha_0 = \theta_0 = 1$.
2. The penalty method is used in [11, 14] to solve the LSE problem (30), but the QR decomposition is used in this paper to solve this problem. The penalty

method requires a parameter $\eta \gg 1$, but a parameter is not required when the QR decomposition is used.

3. It is shown in Section 5.1 that it is necessary to determine the optimal column $\mathbf{c}_q$ of $\mathbf{S}_t(\bar{p}, \alpha_0\bar{q})$, such that the residual of the approximate linear algebraic equation (27) assumes its minimum value with respect to the residuals when the other $r + s - 2t + 1$ columns of $\mathbf{S}_t(\bar{p}, \alpha_0\bar{q})$ are moved to the right hand side. The optimal column of $\mathbf{S}_t(p, q)$ is defined as its first column in [11, 14], but it is shown in [27] that this choice may yield bad results, and that the optimal column of $\mathbf{S}_t(\bar{p}, \alpha_0\bar{q})$ must be determined for each problem.

5.2 Convergence of the LSE problem

This section considers the convergence of the LSE problem (30), which is a problem of the form

$$\min_{\mathbf{y}} \|\mathbf{y} - \mathbf{p}\|_2 \qquad \text{subject to} \qquad \mathbf{D}\mathbf{y} = \mathbf{q},$$

where $\mathbf{D} \in \mathbb{R}^{r\times s}, \mathbf{y}, \mathbf{p} \in \mathbb{R}^s, \mathbf{q} \in \mathbb{R}^r$ and $r < s$.[3] This problem can be solved by the QR decomposition, as shown in Algorithm 1 [7].

Algorithm 1: The solution of the LSE problem by the QR decomposition

(a) Compute the QR decomposition of $\mathbf{D}^T$,

$$\mathbf{D}^T = \mathbf{Q}\mathbf{R} = \mathbf{Q}\begin{bmatrix}\mathbf{R}_1\\ \mathbf{0}\end{bmatrix}.$$

(b) Set $\mathbf{w}_1 = \mathbf{R}_1^{-T}\mathbf{q}$.
(c) Partition $\mathbf{Q}$ into

$$\mathbf{Q} = \begin{bmatrix}\mathbf{Q}_1 \ \mathbf{Q}_2\end{bmatrix}, \tag{33}$$

where $\mathbf{Q}_1 \in \mathbb{R}^{s\times r}$ and $\mathbf{Q}_2 \in \mathbb{R}^{s\times(s-r)}$.
(d) Compute $\mathbf{w}_2 = \mathbf{Q}_2^T\mathbf{p}$.
(e) Compute the solution

$$\mathbf{y} = \mathbf{Q}\begin{bmatrix}\mathbf{w}_1\\ \mathbf{w}_2\end{bmatrix} = \mathbf{Q}\begin{bmatrix}\mathbf{R}_1^{-T}\\ \mathbf{0}\end{bmatrix}\mathbf{q} + \mathbf{Q}_2\mathbf{Q}_2^T\mathbf{p}. \tag{34}$$

Consider the application of this algorithm to the solution of (30). In particular, it follows from (32) and (34) that

$$\delta\mathbf{y}^{(j)} = \mathbf{Q}^{(j)}\begin{bmatrix}\mathbf{R}_1^{-T}\\ \mathbf{0}\end{bmatrix}^{(j)}\mathbf{r}^{(j)} + \left(\mathbf{Q}_2\mathbf{Q}_2^T\right)^{(j)}\left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right),$$

[3] The integers r and s are not related to the degrees of the polynomials $\hat{p}(y)$ and $\hat{q}(y)$, and polynomials derived from them, that are introduced in Section 5.

at the jth iteration. If $\bar{\mathbf{y}}$ is the solution of (30), and $\mathbf{e}^{(j)}$ and $\mathbf{e}^{(j-1)}$ are the errors at the jth and $(j-1)$th iterations, then it follows from (31) that

$$\begin{aligned}\mathbf{e}^{(j)} &= \mathbf{y}^{(j)} - \bar{\mathbf{y}} \\ &= \mathbf{e}^{(j-1)} + \left(\mathbf{Q}_2\mathbf{Q}_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \mathbf{Q}^{(j)} \begin{bmatrix} \mathbf{R}_1^{-T} \\ \mathbf{0} \end{bmatrix}^{(j)} \mathbf{r}^{(j)},\end{aligned}$$

and thus

$$\begin{aligned}\mathbf{e}^{(j)} - \mathbf{e}^{(j-1)} &= \left(\mathbf{Q}_2\mathbf{Q}_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \mathbf{Q}^{(j)} \begin{bmatrix} \mathbf{R}_1^{-T} \\ \mathbf{0} \end{bmatrix}^{(j)} \mathbf{r}^{(j)}, \\ &= \mathbf{Q}^{(j)} \left(\left(\mathbf{Q}^T\right)^{(j)} \left(\mathbf{Q}_2\mathbf{Q}_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \begin{bmatrix} \mathbf{R}_1^{-T} \\ \mathbf{0} \end{bmatrix}^{(j)} \mathbf{r}^{(j)} \right), \\ &= \mathbf{Q}^{(j)} \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_2^T \end{bmatrix}^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \begin{bmatrix} \mathbf{R}_1^{-T} \\ \mathbf{0} \end{bmatrix}^{(j)} \mathbf{r}^{(j)} \right),\end{aligned}$$

from (33) because

$$\mathbf{Q}_1^T\mathbf{Q}_1 = \mathbf{I}_r, \qquad \mathbf{Q}_2^T\mathbf{Q}_2 = \mathbf{I}_{s-r} \qquad \text{and} \qquad \mathbf{Q}_1^T\mathbf{Q}_2 = \mathbf{0}.$$

It follows that the iterative scheme converges if

$$\begin{aligned}\lim_{j\to\infty} \left\|\mathbf{e}^{(j)} - \mathbf{e}^{(j-1)}\right\|_2 &= \lim_{j\to\infty} \left\|\mathbf{y}^{(j)} - \mathbf{y}^{(j-1)}\right\|_2 \\ &= \lim_{j\to\infty} \left\| \begin{bmatrix} \left(\mathbf{R}_1^{-T}\mathbf{r}\right)^{(j)} \\ \left(\mathbf{Q}_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) \end{bmatrix} \right\|_2 = 0,\end{aligned}$$

and thus two conditions must be satisfied for its convergence:

1. It is necessary that $\mathbf{r}^{(j)} \to \mathbf{0}$ as $j \to \infty$.
2. The vector $\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}$ must lie in the nullspace of $\left(\mathbf{Q}_2^T\right)^{(j)}$ as $j \to \infty$.

Many computational experiments showed that (30) converges in fewer than five iterations, even when a large amount of noise is added to the exact image to form a highly blurred image.

6 Extension to a non-separable PSF

It has been shown that a separable PSF can be calculated from one blurred image, and it can then be deconvolved from the blurred image, thereby yielding a deblurred form of the blurred image. It is shown in this section that if the PSF is non-separable, then two blurred images are required for its calculation, and that computations that are not required for a separable PSF must be included for the

computation of a deblurred image that is formed by a non-separable PSF. It is therefore appropriate to consider changes to the method, such that its modified form can be used for the solution of the BID problem when a non-separable PSF is used.

It was shown in Section 4 that the solution of the BID problem requires one blurred image if the PSF is separable, and that the computation of the PSF requires the selection of two rows and two columns from the blurred image $\mathcal{G}$. If, however, the PSF is non-separable, two blurred images $\mathcal{G}_1$ and $\mathcal{G}_2$ are required, and it is assumed, for simplicity, they are the same size, $(M + p) \times (N + r)$ pixels, where the exact images are $M \times N$ pixels and the PSF is $(p+1) \times (r+1)$ pixels. If $r_1(i)$ and $r_2(i)$ are the ith rows of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively, then the algorithm discussed in Section 5 is applied to every pair of rows $(r_1(i), r_2(i)), i = 1, \ldots, M + p$, of $\mathcal{G}_1$ and $\mathcal{G}_2$ when a non-separable PSF is used. The LSE problem (30) is therefore solved for each of these pairs of rows, and thus a deblurred form of each row of $\mathcal{G}_1$ and each row of $\mathcal{G}_2$, and the PSF of each row, are obtained.

This procedure is repeated for the columns $c_1(i)$ and $c_2(i)$, $i = 1, \ldots, N + r$, of, respectively, $\mathcal{G}_1$ and $\mathcal{G}_2$, and thus a deblurred form of each column of $\mathcal{G}_1$, and each column of $\mathcal{G}_2$, and the PSF of each column, are obtained. It follows, therefore, that there are two sets of results, one set obtained by considering the rows of $\mathcal{G}_1$ and $\mathcal{G}_2$, and one set obtained by considering the columns of $\mathcal{G}_1$ and $\mathcal{G}_2$. The deblurred forms of each row and column of $\mathcal{G}_1$ and $\mathcal{G}_2$, and the components of the PSF along each row and column, are obtained by a sequence of independent AGCD computations, and they therefore have scale factors associated with them. These scale factors must be removed in order to compute the deblurred images and the PSF, and a method for their removal is described in [15, 16, 19].

The method discussed above differs from the method used by Pillai and Liang [19] because they assume that only one blurred image $\mathcal{G}$ is available and that the degrees in x and y of the PSF are small. They propose that $\mathcal{G}$ be partitioned into two regions $\mathcal{R}_1$ and $\mathcal{R}_2$, such that each region contains the entire PSF. The assumption of small supports of the PSF is necessary in order to localise the effects of the partition to the region $\mathcal{L}$ of the common edge of $\mathcal{R}_1$ and $\mathcal{R}_2$, such that the region $\mathcal{L}$ is small and the effects of the partition do not propagate into the interiors of $\mathcal{R}_1$ and $\mathcal{R}_2$. There are, however, differences between the work described in this paper and the work in [19], such that the method described in [19] may be limited. As noted above, it is assumed that the supports of the PSF are small, and it is stated in [19] that the method produces reliable results when the SNR is greater than 40 dB, which is much larger than the SNRs of 15.62 dB and 13.14 dB of the blurred images in Examples 1 and 2 in Section 7. Also, it is assumed in [19] that only additive noise $N(x, y)$ is present, and that the uncertainty $E(x, y)$ in the PSF is zero. By contrast, it is assumed in this paper that $N(x, y) \neq 0$ and $E(x, y) \neq 0$, and Example 3 in Section 7 shows that the uncertainty in the PSF is a significantly greater cause of blur than is additive noise.

7 Examples

This section contains examples that show the deblurred images that result from the AGCD computations discussed in Sections 4 and 5, and they are compared with the deblurred images from four other methods. The examples show that the deblurred images obtained from the AGCD computations and polynomial divisions are significantly better than the deblurred images that result from the other methods, even though these other methods require that the PSF be known. This must be compared with the method described in this paper, which allows the PSF to be calculated from the blurred image.

A blurred image is formed by perturbing the PSF and adding random noise, and thus the blurring model used is obtained by including in (1) the uncertainty $\mathcal{E}$ in the PSF,

$$\mathcal{G} = (\mathcal{H} + \mathcal{E}) \otimes \mathcal{F} + \mathcal{N}, \tag{35}$$

which can also be expressed as a generalised form of (2),

$$\mathbf{g} = (\mathbf{H} + \mathbf{E})\mathbf{f} + \mathbf{n}. \tag{36}$$

The PSF that is applied to the exact images in Examples 1, 2 and 3 is shown in Figure 1. It is seen that the row and column components of the PSF are represented by polynomials of degrees 8 and 10 respectively, that the decays of the PSF in the directions of increasing and decreasing row index are not equal, and that the decays of the PSF in the directions of increasing and decreasing column index are not equal.

The deblurred images obtained from the AGCD computations and polynomial divisions discussed in this paper are compared with the deblurred images obtained from the Lucy-Richardson algorithm, blind image deconvolution, a regularised filter and the Wiener filter [9]. The deblurred images from these methods are obtained using the image processing toolbox in MATLAB. The function calls for these methods are now considered.

Lucy-Richardson algorithm The function call is

`[f]=deconvlucy(g,psf,numit,dampar,weight)`

where `f` is the deblurred image, `g` is the blurred image, `psf` is the PSF, `numit` is the number of iterations, `dampar` is a scalar threshold that defines the deviation of the pixel values of the deblurred image from the pixel values of the blurred image, such that iterations are suppressed for pixels that deviate less than `dampar` from their original values, and `weight` is an array, of the same size as `g`, that assigns a weight to each pixel that reflects its quality. The default values `numit`=10 and `dampar`=0, which corresponds to no damping, are used, and `weight` is equal to the matrix, all of whose entries are equal to one. This specification implies that all pixels in `g` have the same error, and all pixels are therefore treated equally in the algorithm.

Blind image deconvolution The function call is

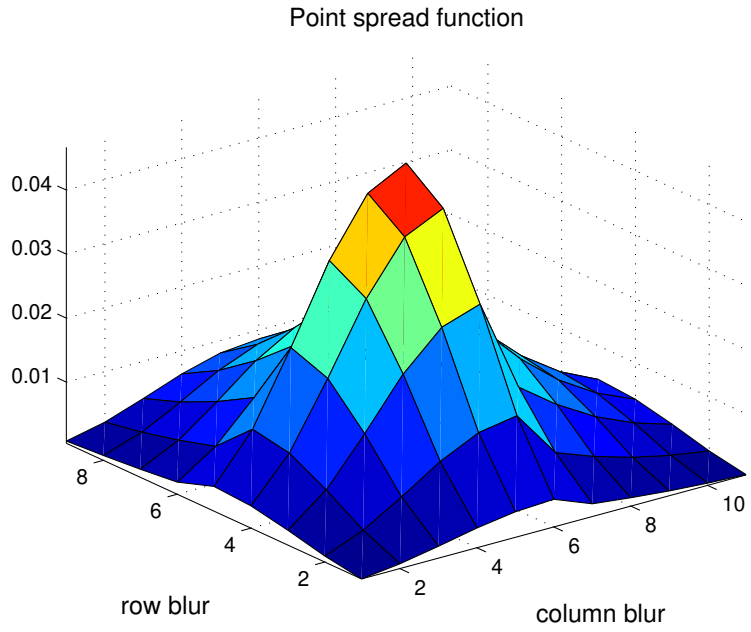


Fig. 1. The PSF that is applied to the exact images in Examples 1, 2 and 3.

```
[f,psf]=deconvblind(g,initpsf,numit,dampar,weight)
```

where `f,g,numit,dampar` and `weight` are defined in the specification of the function call for the Lucy-Richardson algorithm, `initpsf` is an initial estimate of the PSF, and `psf` is an improved estimate of the PSF. The default values of `numit,dampar` and `weight`, which are defined in the specification of the Lucy-Richardson algorithm, are used in Examples 1, 2 and 3, and the parameter `initpsf` is set equal to the exact PSF.
The effect on the deblurred image `f` of the number of iterations `numit` in the Lucy-Richardson algorithm and blind image deconvolution is considered in [8].

Regularised filter The function call is

```
[f]=deconvreg(g,psf,noisepower,range)
```

where `f,g` and `psf` are defined in the specification of the function call for the Lucy-Richardson algorithm, `noisepower` is equal to the noise power and `range` is the range within which the optimal value of a parameter for the satisfaction of a constraint is sought. The default value of `range`, which is equal to $[10^{-9}, 10^{9}]$, is used.

Wiener filter The function call is

```
[f]=deconvwnr(g,psf,r)
```

where `f,g` and `psf` are defined in the specification of the function call for the Lucy-Richardson algorithm. The argument `r` is the ratio of the noise power to the signal power, and it follows from (36) that, assuming $\|\mathbf{E}\| \approx \mathbf{0}$,

$$\mathtt{r} \approx \frac{\|\mathbf{n}\|_2^2}{\|\mathbf{f}\|_2^2}. \tag{37}$$

If the argument `r` is omitted, the function returns an ideal inverse filter.

The default values of the arguments of these functions are used in Examples 1 and 2.

(a) (b)

Fig. 2. (a) An original (exact) image and (b) a blurred image obtained after the addition of uncertainty to the PSF and random noise, for Example 1.

Example 1 Figure 2(a) shows an exact image that is blurred by adding uncertainty $\mathcal{E}$ to the PSF and random noise $\mathcal{N}$, as shown in (35). In particular, let $((\mathcal{H}+\mathcal{E})\otimes\mathcal{F})_{i,j}$, $\mathcal{E}_{i,j}$, $\mathcal{H}_{i,j}$ and $\mathcal{N}_{i,j}$ denote entries (i,j) of $(\mathcal{H}+\mathcal{E})\otimes\mathcal{F}$, $\mathcal{E}$, $\mathcal{H}$ and $\mathcal{N}$ respectively. These entries satisfy

$$0 < \frac{\mathcal{E}_{i,j}}{\mathcal{H}_{i,j}} \leq 10^{-5}, \qquad i = 0,\ldots,p; j = 0,\ldots,r, \tag{38}$$

and

$$0 < \frac{\mathcal{N}_{i,j}}{\left((\mathcal{H}+\mathcal{E})\otimes\mathcal{F}\right)_{i,j}} \leq 10^{-5}, \qquad i = 0,\ldots,M+p-1; j = 0,\ldots,N+r-1, \tag{39}$$

and the values of $\mathcal{E}_{i,j}$ and $\mathcal{N}_{i,j}$ yield a SNR of

$$20 \log_{10} \frac{\|\mathbf{f}\|_2}{\|\mathbf{g} - \mathbf{f}\|_2} = 15.62 \text{ dB}, \tag{40}$$

where $\mathbf{f}$ and $\mathbf{g}$ are defined in (36). The uncertainty $\mathcal{E}$ and noise $\mathcal{N}$ were used to blur the image, thereby obtaining the blurred image shown in Figure 2(b). The AGCD computations and polynomial divisions discussed in Sections 4 and 5 were then used to deblur this image, and the deblurred image is shown in Figure 3. The relative errors in (38) and (39) were not used in the algorithm for the computation of the deblurred image in Figure 3, as discussed in Section 1.

Fig. 3. The deblurred image obtained by AGCD computations and polynomial divisions for Example 1.

This deblurred image was compared with the deblurred images computed by the Lucy-Richardson algorithm, blind image deconvolution, a regularised filter and the Wiener filter. The results are shown in Figures 4 and 5, and it is clear that the best deblurred image is obtained using AGCD computations and polynomial divisions. It is important to note that the deblurred images in Figures 4 and 5 were obtained by specifying the exact PSF, but the PSF was calculated in order to obtain the deblurred image in Figure 3.

Quantitative comparison of the exact image and the deblurred images in Figures 3, 4 and 5 requires care because the deblurred image in Figure 3 is obtained by AGCD computations and polynomial divisions. In particular, it follows from (24) that an AGCD is defined to within an arbitrary non-zero scalar multiplier, and since the computed AGCD is equal to the PSF, which is deconvolved from the blurred image, it follows that the deblurred image from AGCD computations

and polynomial divisions is also defined to within an arbitrary non-zero scalar multiplier. Comparison of the exact image with the deblurred images in Figures 3, 4 and 5 requires, therefore, that all the images be normalised.

If $\hat{\mathbf{F}}$ and $\bar{\mathbf{F}}$ are, respectively, the normalised forms of the matrix representations of the exact image $\mathcal{F}$ and a deblurred image $\ddot{\mathcal{F}}$ whose matrix representations are $\mathbf{F}$ and $\ddot{\mathbf{F}}$ respectively, then

$$\hat{\mathbf{F}} = \frac{\mathbf{F}}{\|\mathbf{F}\|_2} \qquad \text{and} \qquad \bar{\mathbf{F}} = \frac{\ddot{\mathbf{F}}}{\left\|\ddot{\mathbf{F}}\right\|_2}.$$

The SNR between the exact image and a deblurred image is therefore

$$\mu = 20 \log_{10} \frac{\left\|\hat{\mathbf{F}}\right\|_F}{\left\|\hat{\mathbf{F}} - \bar{\mathbf{F}}\right\|_F} \text{ dB}, \tag{41}$$

where the subscript F denotes the Frobenius norm. Table 1 shows the values of μ for the deblurred images in Figures 3, 4 and 5, and it is seen that the maximum value of μ occurs for the deblurred image obtained by AGCD computations and polynomial divisions, which is evident from the deblurred images in these figures.

It follows from Table 1 and (40) that μ is approximately equal to the SNR of the given blurred image $\mathcal{G}$ for blind image deconvolution, the Wiener filter and the Lucy-Richardson algorithm, which shows their regularisation property. The deblurred image from the regularised filter is obtained by the least squares minimisation of the error between the estimated image and the true image, with a constraint on the preservation of the smoothness of the image. The effect of different values of the arguments `noisepower` and `range` on the deblurred image obtained from the function `deconvreg.m` is considered in [8] and it is noted that experiments may be needed to determine the values of these parameters that yield the best deblurred image. Specifically, an example in [8] shows that the best deblurred image is obtained, by experiment, when the noise power is equal to 10% of its initial estimate and `range` is equal to $\left[10^{-7}, 10^{7}\right]$, which is tighter than the default range.

The deblurred image from the Wiener filter was obtained by including the value of `r`, which is defined in (37), in the arguments of the function `deconvwnr.m`. The value of `r` may not, however, be known in practical problems, but Table 1 shows that even when it is known, the value of μ is approximately equal to the values of μ for the deblurred images from blind image deconvolution and the Lucy-Richardson algorithm. The function `deconvblind.m` that implements blind image deconvolution requires an initial estimate of the PSF, and the function returns a better estimate. In this example, the function `deconvblind.m` was called with the exact PSF, which is known, as shown by the classification of the PSF in Table 1.

An expression for the error in the computed PSF requires that it be normalised so that the sum of the elements of its matrix representation is one. In

particular, if $\hat{\mathbf{H}}$ and $\bar{\mathbf{H}}$ are the matrices of the exact and computed PSFs that are normalised such that their elements, $\hat{h}_{i,j}$ and $\bar{h}_{i,j}$ respectively, satisfy

$$\sum_{i,j} \hat{h}_{i,j} = 1 \qquad \text{and} \qquad \sum_{i,j} \bar{h}_{i,j} = 1,$$

then the error between the computed and exact PSFs is

$$\lambda = \sum_{i,j} \left| \hat{h}_{i,j} - \bar{h}_{i,j} \right|. \tag{42}$$

The error λ between the computed and exact PSFs is $1.42\exp{-05}$. This is approximately equal to the lower bound of the componentwise error in the exact PSF, which is specified in (38). □

(a) (b)

Fig. 4. Deblurred images of the image in Figure 2(b) obtained by (a) the Lucy-Richardson algorithm and (b) blind image deconvolution, for Example 1.

Example 2 The procedure described in Example 1 was repeated for the exact image shown in Figure 6(a), and the blurred image shown in Figure 6(b) was obtained by perturbing the PSF and adding random noise, as shown in (38) and (39) respectively. The SNR of the blurred image is 13.14 dB.

The deblurred image obtained by AGCD computations and polynomial divisions is shown in Figure 7, and the deblurred images obtained using the Lucy-Richardson algorithm, blind image deconvolution, a regularised filter and the

(a) (b)

Fig. 5. Deblurred images of the image in Figure 2(b) obtained by (a) a regularised filter and (b) the Wiener filter, for Example 1.

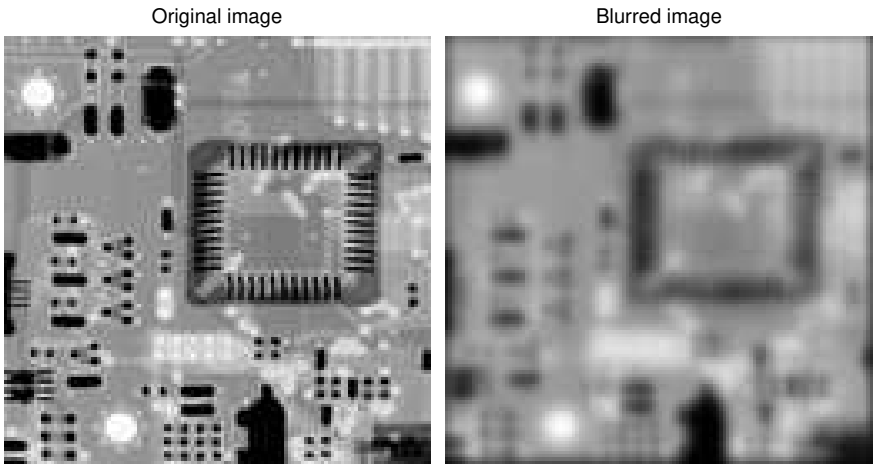
Method	PSF known/not known	μ (dB)
Blind image deconvolution	known	15.26
Regularised filter	known	9.19
Wiener filter	known	15.67
Lucy-Richardson	known	15.16
AGCDs and poly. divisions	not known (calculated)	62.63

Table 1. The SNRs of the deblurred images for Example 1.

Wiener filter are shown in Figures 8 and 9. Table 2 shows the value of the SNR, which is defined in (41), for each deblurred image in Figures 7, 8 and 9, and it is seen that the results of Example 2 are very similar to the results of Example 1 because the deblurred image obtained from the AGCD computations and polynomial divisions has the largest value of μ.

The error between the computed and exact PSFs, using the error measure (42), is $1.41 \exp -05$. This is approximately equal to the lower bound of the componentwise error in the exact PSF, which is defined in (38). □

Example 3 The blur in Examples 1 and 2 is caused by the uncertainty $\mathcal{E}$ in the PSF $\mathcal{H}$, and additive noise $\mathcal{N}$. The relative importance of these sources of blur was investigated by performing three experiments on the exact image shown in Figure 6(a):



(a) (b)

Fig. 6. (a) An original (exact) image and (b) a blurred image obtained after the addition of uncertainty to the PSF and random noise, for Example 2.

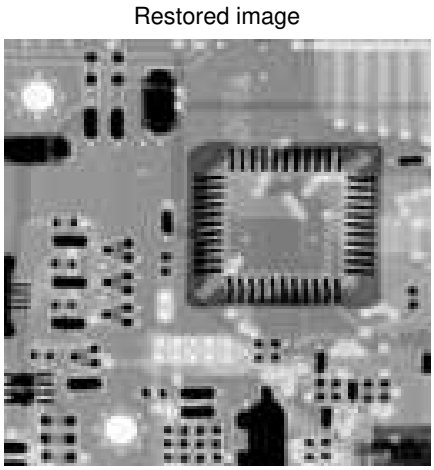
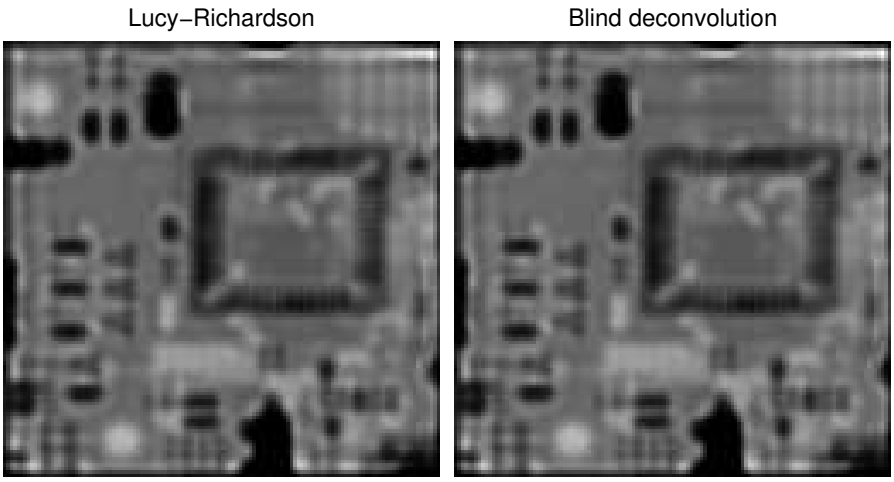


Fig. 7. The deblurred image obtained by AGCD computations and polynomial divisions for Example 2.

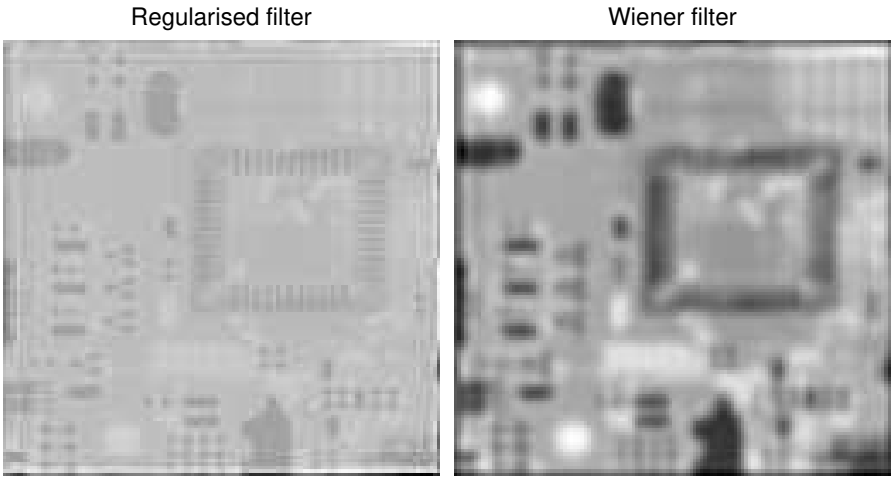
Experiment 1: The uncertainty in the PSF is zero, $\mathcal{E} = 0$, and the pixel value $\mathcal{N}(i,j)$ of the noise $\mathcal{N}$ satisfies

$$0 < \frac{\mathcal{N}(i,j)}{\mathcal{S}(i,j)} \leq \epsilon, \qquad \mathcal{S} = (\mathcal{H} + \mathcal{E}) \otimes \mathcal{F}, \qquad \epsilon = 10^{-10}, 10^{-9}, \ldots, 10^{-3},$$



(a) (b)

Fig. 8. Deblurred images of the image in Figure 6(b) obtained by (a) the Lucy-Richardson algorithm and (b) blind image deconvolution, for Example 2.



(a) (b)

Fig. 9. Deblurred images of the image in Figure 6(b) obtained by (a) a regularised filter and (b) the Wiener filter, for Example 2.

where $i = 0, \ldots, M + p - 1$, and $j = 0, \ldots, N + r - 1$.

Method	PSF known/not known	μ (dB)
Blind image deconvolution	known	13.12
Regularised filter	known	7.94
Wiener filter	known	13.72
Lucy-Richardson	known	12.96
AGCDs and poly. divisions	not known (calculated)	62.64

Table 2. The SNRs of the deblurred images for Example 2.

Experiment 2: The additive noise is zero, $\mathcal{N} = 0$, and the pixel value $\mathcal{E}(k,l)$ of the uncertainty $\mathcal{E}$ in the PSF $\mathcal{H}$ satisfies

$$0 < \frac{\mathcal{E}(k,l)}{\mathcal{H}(k,l)} \leq \epsilon, \qquad \epsilon = 10^{-10}, 10^{-9}, \ldots, 10^{-3},$$

for $k = 0, \ldots, p$, and $l = 0, \ldots, r$.

Experiment 3: The uncertainty $\mathcal{E}$ in the PSF and additive noise $\mathcal{N}$ satisfy

$$0 < \frac{\mathcal{E}(k,l)}{\mathcal{H}(k,l)}, \frac{\mathcal{N}(i,j)}{\mathcal{S}(i,j)} \leq \epsilon, \qquad \epsilon = 10^{-10}, 10^{-9}, \ldots, 10^{-3}.$$

Figure 10 shows the variation of the relative error γ between the exact and deblurred images with the relative error ϵ. The results for Experiments 2 and 3 are almost identical and they are therefore shown together. The graphs show that $\gamma = \epsilon$ for Experiment 1, and $\gamma = 100\epsilon$ for Experiments 2 and 3, and thus the relative error in a deblurred image is dominated by uncertainty in the PSF, and the effect of additive noise is relatively small. □

The Bézout matrix, rather than the Sylvester matrix, is used in [15] for the solution of the BID problem. The Bézout matrix has half the number of rows and half the number of columns of the Sylvester matrix if the polynomials are of the same degree, and it is symmetric, and both properties suggest it is advantageous to perform the AGCD computations using this matrix, rather than the Sylvester matrix. The disadvantage of the Bézout matrix is the requirement to evaluate terms of the form $p_i q_j - p_j q_i$ for its formation, and thus numerical problems may arise because of cancellation. Also, the SNR of the blurred images in the examples in [15] is greater than 50 dB, and this value is much larger than the SNRs of the blurred images in Examples 1 and 2, which are 15.62 dB and 13.14 dB respectively.

Equation (28) arises because the addition of two Sylvester matrices is also a Sylvester matrix, assuming the degrees of the polynomials are consistent. This equation does not apply to the Bézout matrix $\mathbf{B}(f,g)$ because the addition of two Bézout matrices does not yield a Bézout matrix since each entry is a bilinear function of the coefficients of $f = f(y)$ and $g = g(y)$.

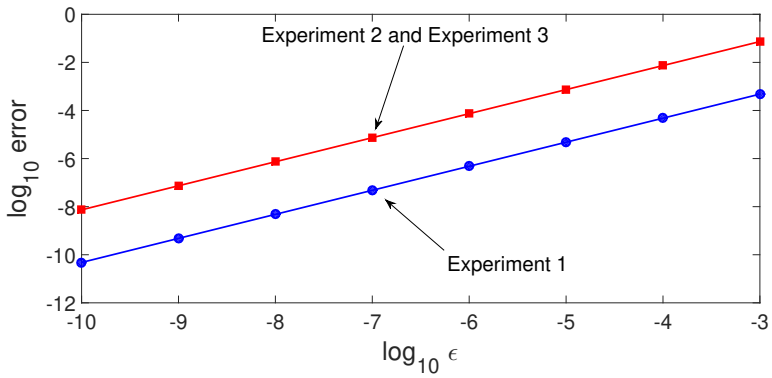


Fig. 10. The relative contributions of the additive noise and uncertainty in the PSF to the error in a deblurred image, for Experiment 1, and Experiments 2 and 3.

8 Summary

This paper has considered the application of AGCD computations and polynomial divisions to the removal of blur from an image. These polynomial operations can be used if a blurred image is formed by the convolution of the exact image and a spatially invariant PSF because the multiplication of two polynomials reduces to the convolution of their coefficients if this condition on the PSF is satisfied. The deblurred image obtained from these polynomial computations was compared with the deblurred images obtained from four other methods, and the deblurred image of highest quality was obtained from the method that uses polynomial computations.

The method of deblurring discussed in this paper can be extended to a non-separable PSF, which requires that two blurred images be specified for its calculation. More computations of the same type as occur when a separable PSF is used, and additional computations, are required when the PSF is non-separable. It is also necessary to remove arbitrary scale factors that appear in the rows and columns of the deblurred images when a non-separable PSF is used.

References

1. S. Barnett. *Polynomials and Linear Control Systems.* Marcel Dekker, New York, USA, 1983.
2. P. Chin and R. M. Corless. Optimization strategies for the approximate GCD problem. In *Proc. Int. Symp. Symbolic and Algebraic Computation*, pages 228–235, Rostock, Germany, 1998.
3. R. M. Corless, S. M. Watt and L. Zhi. QR factoring to compute the GCD of univariate approximate polynomials. *IEEE Trans. Signal Processing*, 52(12):3394–3402, 2004.

4. A. Cornelio, E. Piccolomini and J. Nagy. Constrained numerical optimization methods for blind deconvolution. *Numer. Algor.*, 65:23–42, 2014.
5. A. Danelakis, M. Mitrouli and D. Triantafyllou. Blind image deconvolution using a banded matrix method. *Numer. Algor.*, 64:43–72, 2013.
6. D. Fulkerson and P. Wolfe. An algorithm for scaling matrices. *SIAM Review*, 4:142–146, 1962.
7. G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, USA, 2013.
8. R. C. Gonzalez, R. E. Woods and S. L. Eddins. *Digital Image Processing Using* MATLAB. Gatesmark Publishing, 2009.
9. B. Gunturk and X. Li. *Image Registration: Fundamentals and Advanves*. CRC Press, Florida, USA, 2013.
10. P. C. Hansen, J. G. Nagy and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, Philadelphia, USA, 2006.
11. E. Kaltofen, Z. Yang and L. Zhi. Structured low rank approximation of a Sylvester matrix. In D. Wang and L. Zhi, editors, *Trends in Mathematics*, pages 69–83. Birkhäuser Verlag, Basel, Switzerland, 2006.
12. N. K. Karmarkar and Y. N. Lakshman. On approximate GCDs of univariate polynomials. *Journal of Symbolic Computation*, 26:653–666, 1998.
13. D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, 1996.
14. B. Li, Z. Yang and L. Zhi. Fast low rank approximation of a Sylvester matrix by structured total least norm. *J. Japan Soc. Symbolic and Algebraic Comp.*, 11:165–174, 2005.
15. Z. Li, Z. Yang and L. Zhi. Blind image deconvolution via fast approximate GCD. *Proc. Int. Symp. Symbolic and Algebraic Computation*, pages 155–162, 2010.
16. B. Liang and S. Pillai. Blind image deconvolution using a robust 2-D GCD approach. *IEEE Int. Symp. Circuits and Systems*, pages 1185–1188, 1997.
17. J. Nagy, K. Palmer and L. Perrone. Iterative methods for image deblurring: a Matlab object-oriented approach. *Numer. Algor.*, 36:73–93, 2004.
18. S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, USA, 1996.
19. S. Pillai and B. Liang. Blind image deconvolution using a robust GCD approach. *IEEE Trans. Image Processing*, 8(2):295–301, 1999.
20. P. Premaratne and C. Ko. Retrieval of symmetrical image blur using zero sheets. *IEE Proceedings: Vision, Image and Signal Processing*, 148(1):65–69, 2001.
21. J. Ben Rosen, H. Park and J. Glick. Total least norm formulation and solution for structured problems. *SIAM J. Mat. Anal. Appl.*, 17(1):110–128, 1996.
22. J. Ben Rosen, H. Park and J. Glick. Structured total least norm for nonlinear problems. *SIAM J. Mat. Anal. Appl.*, 20(1):14–30, 1998.
23. B. L. Satherley and C. R. Parker. Two-dimensional image reconstruction from zero sheets. *Optics Letters*, 18:2053–2055, 1993.
24. D. Triantafyllou and M. Mitrouli. Two resultant based methods computing the greatest common divisor of two polynomials. *LNCS*, 3401:519–526, 2005.
25. J. R. Winkler. Polynomial computations for blind image deconvolution, 2015. Submitted.
26. 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*, 234:3226–3242, 2010.

27. J. R. Winkler and M. Hasan. An improved non-linear method for the computation of a structured low rank approximation of the Sylvester resultant matrix. *Journal of Computational and Applied Mathematics*, 237(1):253–268, 2013.
28. J. R. Winkler, M. Hasan and X. Y. Lao. Two methods for the calculation of the degree of an approximate greatest common divsior of two inexact polynomials. *Calcolo*, 49:241–267, 2012.
29. C. J. Zarowski, X. Ma and F. W. Fairman. QR-factorization method for computing the greatest common divisor of polynomials with inexact coefficients. *IEEE Trans. Signal Processing*, 48(11):3042–3051, 2000.
30. Z. Zeng. The approximate GCD of inexact polynomials. Part 1: A univariate algorithm, 2004. Preprint.