Article:
Winkler, J.R. (2016) Polynomial computations for blind image deconvolution. Linear Algebra and Its Applications, 502. pp. 77-103. ISSN 0024-3795

https://doi.org/10.1016/j.laa.2015.10.010

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Polynomial computations for blind image deconvolution

Joab R. Winkler

Department of Computer Science, The University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom

ARTICLE INFO

Article history:
Received 24 January 2015
Accepted 11 October 2015
Available online xxxx
Submitted by V. Mehrmann

MSC:
68U10
12Y05

Keywords:
Blind image deconvolution
Sylvester resultant matrix
Structured matrices

ABSTRACT

This paper considers the problem of blind image deconvolution (BID) when the blur arises from a spatially invariant point spread function (PSF) $\mathcal{H}$, which implies that a blurred image $\mathcal{G}$ is formed by the convolution of $\mathcal{H}$ and the exact form $\mathcal{F}$ of $\mathcal{G}$. Since the multiplication of two bivariate polynomials is performed by convolving their coefficient matrices, the equivalence of the formation of a blurred image and the product of two bivariate polynomials implies that BID can be performed by considering $\mathcal{F}$, $\mathcal{G}$ and $\mathcal{H}$ to be bivariate polynomials on which polynomial operations are performed. These operations allow the PSF to be computed, which is then deconvolved from the blurred image $\mathcal{G}$, thereby obtaining a deblurred image that is a good approximation of the exact image $\mathcal{F}$. Computational results show that the deblurred image obtained using polynomial computations is better than the deblurred image obtained using other methods for blind image deconvolution.

E-mail address: j.r.winkler@sheffield.ac.uk.

http://dx.doi.org/10.1016/j.laa.2015.10.010
0024-3795/© 2015 Published by Elsevier Inc.

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010

1. Introduction

The removal of blur from an image is an important problem that has many applications, including the diagnosis of medical conditions, the analysis of astronomical data, and security and surveillance. Image interrogation for, for example, feature extraction and segmentation, is easier if the image on which these operations are performed is of high quality, and thus image deblurring can be regarded as a preprocessing operation for subsequent image interrogation. The importance of this preprocessing stage, the many applications in which blurred images arise and the ill-conditioned nature of the problem provide the motivation for the continued development of numerically stable methods for the removal of blur from an image.

The point spread function (PSF) models the blur, and if this function is spatially invariant, which is assumed in this paper, then a blurred image is formed by the convolution of the PSF and the exact image. The problem of blind image deconvolution (BID) is characterised by partial knowledge, or the absence of knowledge, of the PSF, in which case BID is difficult because it reduces to the separation of two convolved signals, either or both of which are not fully specified. This paper describes a method for the solution of the problem of BID in which the blurred and deblurred images, and the PSF, are considered to be bivariate polynomials whose coefficients are the pixel values. Polynomial operations, in particular, approximate greatest common divisor (AGCD) computations and deconvolutions, are used to compute a deblurred image, and it is shown that the deblurred images obtained from this method are of higher quality than the deblurred images obtained from other methods.

There exist several methods for the solution of the BID problem, including probabilistic methods [6,18], autoregressive moving average parameter estimation methods [19] and zero sheet separation [20,29]. Carasso [7] uses Fourier techniques to deblur an image, but the PSF and blurred image must satisfy specified conditions. The work described in this paper uses the Sylvester resultant matrix[1] [2] to deblur an image, and it is therefore similar to the deblurring methods described in [12,22,25], but there are two significant differences between the work described in this paper and other work:

- In this work, a blurred image is formed by the addition of noise to the exact image, such that the signal-to-noise ratio (SNR) spans one order of magnitude across the blurred image. This feature is included because it cannot be assumed in practical image deblurring problems that the SNR is constant across the blurred image, and furthermore, the lower and upper bounds of the SNR may be known approximately and not exactly. These features cause difficulties in deblurring algorithms that involve an iterative procedure in which the termination criterion requires the comparison of the norm of an error vector against a given scalar threshold that is a measure of the SNR. In particular, the specification of this threshold for deblurring an image

[1] This matrix will, for brevity, be termed the Sylvester matrix in the sequel.

in which the SNR varies across the image is difficult because the deblurred image will contain significant noise if the threshold is too small, and significant parts of the exact image will be absent from the deblurred image if the threshold is too high. It is, however, shown in this paper that a high quality deblurred image can be obtained, even if the SNR is low and spans one order of magnitude across the blurred image.

- Structured matrices, for example, the Kronecker product of two matrices that have Tœplitz and Hankel forms [5], the Sylvester matrix [12,22,25] and the Bézout matrix [13,21], arise in the solution of the BID problem when the PSF is spatially invariant. Also, the penalty method is used in [24,26] to solve a constrained minimisation problem in which the coefficient matrix has structure and the constraint is a regulariser that controls the magnitude or smoothness of the pixel values of the deblurred image. This paper uses the method of structured non-linear total least norm (SNTLN) [28], which is a method for the preservation of a non-linear structure in a matrix, to solve a constrained minimisation problem in which the coefficient matrix in the objective function is derived from the Sylvester matrix. It is shown that a non-linear method, rather than a linear method, allows the introduction of two parameters that can be optimised to yield very good results for the AGCD computations, which are important for the determination of the PSF. A non-linear structure-preserving matrix method has not been used for the solution of the BID problem, and it defines the second difference between this work and other work.

It is assumed in this paper that the PSF is separable, and it is shown in [22,25,32] that the extension of the work described in this paper from a separable PSF to a non-separable PSF requires more computations of the same type as are required for a separable PSF, and extra computations that are not required when a separable PSF is used.

A parameter study of four functions in Matlab that implement BID is considered in Section 2, and it is shown in Section 3 that the convolution operation defines the formation of a blurred image by a spatially invariant PSF, and the multiplication of two polynomials. This allows the problem of BID to be posed in terms of polynomial computations in which the exact image, blurred image and PSF are written as bivariate polynomials. The application of the Sylvester matrix to the computation of an AGCD of two polynomials is considered in Section 4, and Section 5 contains results that show the effectiveness of polynomial methods for image deblurring. Section 6 contains a summary of the paper.

2. Other methods for blind image deconvolution

A blurred image $\mathcal{G}$ that is formed by the convolution of its exact form $\mathcal{F}$ and a spatially invariant PSF $\mathcal{H}$, in the presence of additive noise $\mathcal{N}$, is given by

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

where $*$ denotes convolution. This section considers the four methods for the solution of this equation that are implemented in the image processing toolbox in Matlab. These methods and their functions in Matlab are [17]:

- Blind image deconvolution, `deconvblind.m`
  This function maximises the likelihood that the convolution of the computed PSF and deblurred image is an instance of the blurred image, assuming Poisson noise statistics. It uses an accelerated, damped procedure that is similar to the procedure implemented in the function `deconvlucy.m`. The procedure is iterative, and the function requires an initial estimate of the PSF and returns an improved estimate of the PSF.
- The Lucy–Richardson algorithm, `deconvlucy.m`
  This function implements an accelerated, damped form of the Lucy–Richardson algorithm. This algorithm is iterative, assumes Poisson noise statistics and requires that the PSF be known.
- Regularisation, `deconvreg.m`
  This function performs a constrained least squares minimisation in which the Laplacian is the default regularisation operator. The function requires that the noise power and PSF be known.
- Wiener filter, `deconvwnr.m`
  This function uses a minimum mean-squares error criterion to reduce the noise in the blurred image. It is assumed the PSF and the ratio of the noise power to the signal power (PNSR) are known, but the function implements an ideal inverse filter if the PNSR is omitted from the function call.

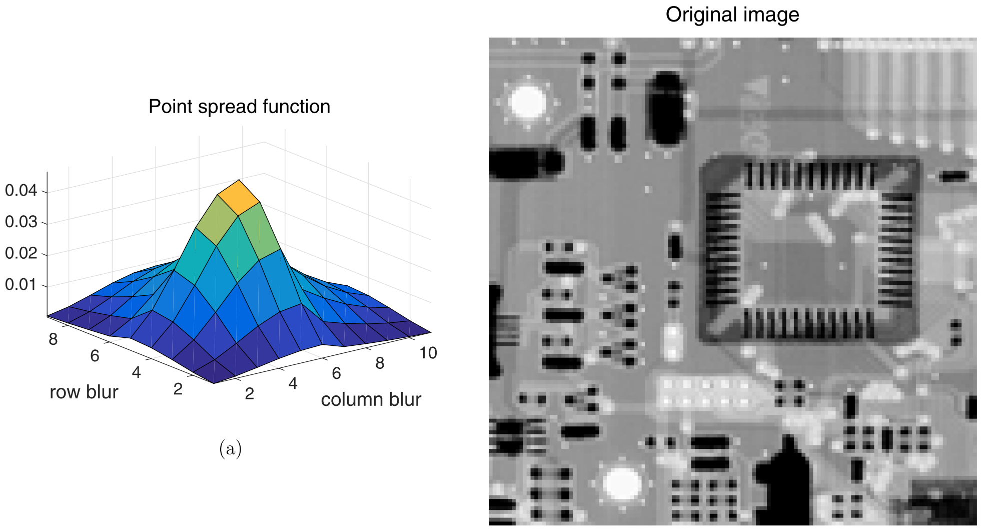
The function `deconvblind.m` implements BID because only an estimate, and not the exact form, of the PSF need be specified in the function call. The other three functions implement, however, linear deconvolution because the exact form of the PSF must be included in their function calls.

The functions `deconvreg.m` and `deconvwnr.m` require that the noise power be known in order to obtain a deblurred image. If F and G are the matrix forms, of order $M \times N$, of the exact and blurred images, then the error matrix is $P = G - F$, and it is required to calculate the power of the image $\mathcal{P}$ represented by this matrix. In particular, the mean μ and the standard deviation σ of each entry $P_{i,j}$ of P are, respectively,

$$\mu = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} P_{i,j}}{MN} \quad \text{and} \quad \sigma = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} Q_{i,j}^2}{MN-1}}, \quad Q_{i,j} = P_{i,j} - \mu,$$

and the average value of the power of each entry $P_{i,j}$ is

$$\mathcal{E}\left\{P_{i,j}^2\right\} = \mathcal{E}\left\{\left((P_{i,j}-\mu)+\mu\right)^2\right\} = \mathcal{E}\left\{(P_{i,j}-\mu)^2\right\} + \mu^2 = \sigma^2 + \mu^2.$$



(a)

(b)

Fig. 1. (a) The PSF that is applied to the image (b).

An estimate of the noise power of the error image $\mathcal{P}$ is therefore

$$MN(\sigma^2 + \mu^2) \approx \|G - F\|_F\,, \tag{2}$$

where the subscript F denotes the Frobenius norm and $\|G - F\|_F$ is the exact value of the noise power in $\mathcal{P}$.

Example 2.1 considers some properties of the four deblurring functions in Matlab.

Example 2.1. Fig. 1(a) shows the PSF $\mathcal{H}$ that is applied to the exact image in Fig. 1(b). The image is $M \times N$ pixels, where $M = N = 128$, and the column and row components of the PSF extend over $p + 1 = 11$ and $q + 1 = 9$ pixels respectively. If the pixel values of $\mathcal{H} * \mathcal{F}$ and the noise $\mathcal{N}$ are $(\mathcal{H} * \mathcal{F})_{i,j}$ and $\mathcal{N}_{i,j}$ respectively, then the noise $\mathcal{N}_{i,j}$ was chosen to satisfy

$$\mathcal{N}_{i,j} \leq \varepsilon\, r_{i,j} (\mathcal{H} * \mathcal{F})_{i,j}, \quad i = 0, \ldots, M + p - 1, \quad j = 0, \ldots, N + q - 1,$$

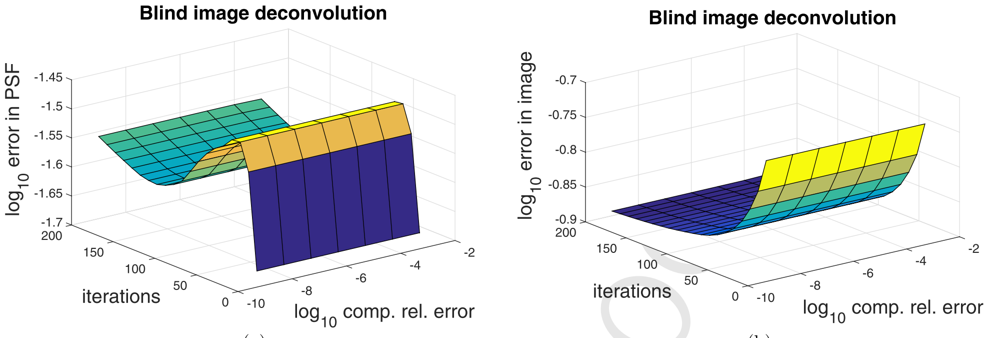
where $r_{i,j}$ is a uniformly distributed random number in the interval $[0, 1]$ and ε is the upper bound of the componentwise relative error. The experiments were repeated for seven values of ε,

$$\left\{10^{-9}, \quad 10^{-8}, \quad 10^{-7}, \quad 10^{-6}, \quad 10^{-5}, \quad 10^{-4}, \quad 10^{-3}\right\}, \tag{3}$$

and thus a blurred image $\mathcal{G}$ was obtained for each value of ε.

The power of the image in Fig. 1(b) is $\|F\|_F^2 = 3.50 \times 10^8$, and although the componentwise relative error ε spans six orders of magnitude, the noise power and the square

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010



(a) (b)

Fig. 2. The relative error in (a) the PSF and (b) the deblurred image, as a function of the number of iterations and the componentwise relative error ε, from `deconvblind.m`.

of the normwise relative error of the blurred images $\mathcal{G}$ are approximately independent of ε,

$$\begin{aligned} \|G - F\|_F^2 &\approx 1.70 \times 10^7, \qquad \varepsilon = 10^{-9},\, 10^{-8}, \ldots, 10^{-4},\, 10^{-3}, \\ \frac{\|G - F\|_F^2}{\|F\|_F^2} &\approx 4.86 \times 10^{-2}, \qquad \varepsilon = 10^{-9},\, 10^{-8}, \ldots, 10^{-4},\, 10^{-3}. \end{aligned} \tag{4}$$

Blurred images display ringing effects at their boundaries, and the function `edgetaper.m` was therefore applied to the blurred images before the four deconvolution functions were called [17]. The function call is

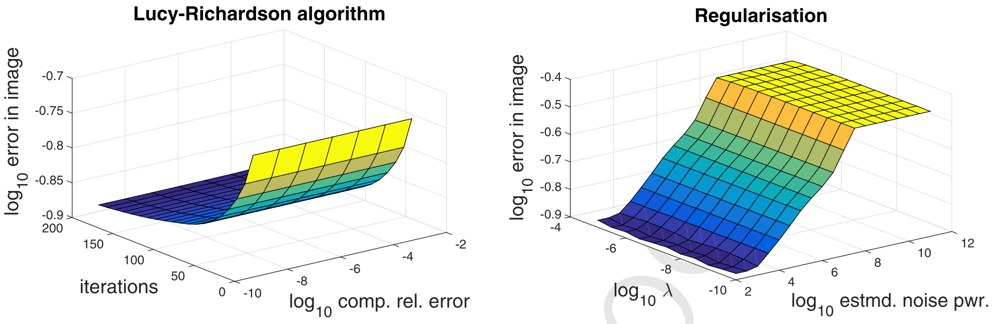
```
[EDGEI] = edgetaper(I,PSF),
```

where `I` and `EDGEI` are blurred images, but `EDGEI` has less ringing than `I` along the borders. The PSF is specified in the argument `PSF`, and its inclusion necessarily implies that the function `edgetaper.m` can only be used when the linear deconvolution problem is solved. It follows that the exact PSF, and not an estimate of the PSF, is specified in the call to the function `deconvblind.m`.

The function call to `deconvblind.m` is

```
[J,PSF] = deconvblind(EDGEI,INITPSF,NUMIT),
```

where `EDGEI` is defined in the function `edgetaper.m` and `J` is the deblurred image, `INITPSF` and `PSF` are the initial and final estimates of the PSF, `NUMIT` is the number of iterations required, and the default values of the other arguments were used. Figs. 2(a) and (b) show, respectively, the variation of the relative error in the PSF and the deblurred image, as a function of the number of iterations and the componentwise relative error ε. The figures show that as the number of iterations increases, the error in the PSF may



(a)

(b)

Fig. 3. (a) The relative error in the deblurred image from (a) `deconvlucy.m`, as a function of the number of iterations and the componentwise relative error ε, and (b) `deconvreg.m`, as a function of the lower bound λ of the Lagrange multiplier and the estimated noise power.

increase or decrease, but the error in the deblurred image decreases. Furthermore, the errors in the PSF and deblurred image are independent of ε.

The function call to `deconvlucy.m` is

$$\texttt{[J] = deconvlucy(EDGEI,PSF,NUMIT)},$$

where `EDGEI`, `J` and `NUMIT` are defined above, `PSF` is the exact PSF, and the default values of the other parameters were used. Fig. 3(a) shows the variation of the relative error in the deblurred image with the number of iterations and the componentwise relative error ε. It is seen that the error is independent of ε and that the error in the deblurred image decreases as the number of iterations increases. Also, Figs. 2(b) and 3(a) are very similar, which is expected.

The function call to `deconvreg.m` is

$$\texttt{[J] = deconvreg(EDGEI,PSF,NP,LRANGE)},$$

where `EDGEI`, `J` and `PSF` are defined above, `NP` is the noise power, `LRANGE` is the range of the Lagrange multiplier within which an optimal solution is sought, and the default values of the other parameters were used. The noise power is estimated from (2) and `LRANGE` is a vector of length two whose entries, λ and λ^{-1}, $\lambda < 1$, are the lower and upper bounds of the range of the Lagrange multiplier. A deblurred image was obtained for eleven values of λ,

$$\texttt{LRANGE} = \begin{bmatrix} \lambda & \lambda^{-1} \end{bmatrix}, \quad \log_{10}\lambda = -10, -9.5, -9.0, \ldots, -6.0, -5.5, -5.0. \tag{5}$$

The aim of the experiment was the determination of the dependence of the error in the deblurred image with the accuracy with which the noise power of the blurred image is known. Fig. 3(b) shows the variation of the relative error in the deblurred image as a function of the estimated noise power and λ, which is defined in (5). Equation (4) shows

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010

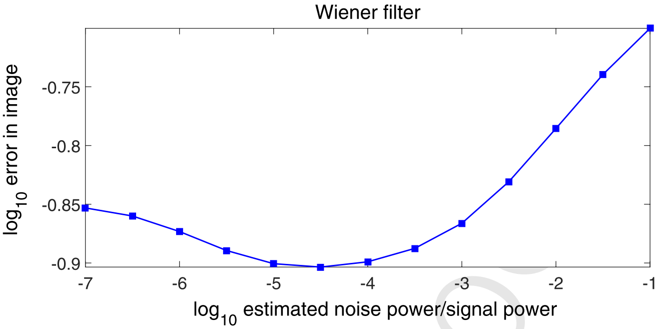


Fig. 4. The error in the deblurred image as a function of the estimated PNSR for the Wiener filter, `deconvwnr.m`.

that the noise power of the blurred image is about 1.70×10^7 for all values of ε, but Fig. 3(b) shows that the noise power `NP` specified in the function call to `deconvreg.m` that yields the smallest error in the deblurred image is about 1000, which is several orders of magnitude smaller than the true noise power. Also, the error in the deblurred image is independent of the range (5) of the Lagrange multiplier, but this result does not imply that the visual appearance of the deblurred images is the same for all the values of λ defined in (5). More generally, computational experiments must be performed to determine the values of the parameters that return the best deblurred image. This is consistent with an example in [17] because the best deblurred image, using visual inspection, is obtained when the noise power specified in the function call is 10% of its true value and the optimal value of λ is $100\lambda_0$, where $\lambda_0 = 10^{-9}$ is the default value of λ.

The function call to `deconvwnr.m` is

$$\texttt{[J] = deconvwnr(EDGEI,PSF,NSR)},$$

where `EDGEI` and `J` are defined above and `NSR` is the estimated value of the PNSR. Fig. 4 shows the variation of the relative error in the deblurred image with the estimated value of the PNSR of the blurred image. Equation (4) shows that PNSR $= 4.86 \times 10^{-2}$, but Fig. 4 shows that the best deblurred image is obtained when PNSR $= 10^{-4.5} = 3.16 \times 10^{-5}$, which is about three orders of magnitude smaller than the true value of the PNSR. □

Figs. 2(b), 3 and 4 show that, using the optimal values of the parameters for each method, the relative error of each deblurred image obtained by the four methods is about $10^{-0.9} = 0.13$ for all the values of ε specified in (3). It follows from (4) that the relative error of each blurred image is $(4.86 \times 10^{-2})^{\frac{1}{2}} = 0.22$, which is about twice the relative error of the deblurred images. Fig. 3(b) shows that a change of about five orders of magnitude in the estimated noise power causes a change of less than one order of magnitude in the error in the deblurred image from `deconvreg.m`, and Fig. 4 shows that a change of six orders of magnitude in the estimated value of the PNSR causes a change of less than one order of magnitude in the error in the deblurred image from deconvwnr.m. The example in Section 5 shows, however, that the specification of an incorrect value of NP in deconvreg.m, and the specification of an incorrect value of NSR in deconvwnr.m, lead to deblurred images that have large errors.

Example 2.1 shows that different values of the arguments of the four functions in Matlab must be considered in order to obtain the best deblurred image, and that it may be necessary to use visual inspection to compare different deblurred images. Finally, the omission of the function edgetaper.m before the four functions were called caused an increase in the errors in the deblurred images, and the estimated PSF from the function deconvblind.m, which shows its effectiveness in reducing ringing at the borders of a blurred image.

3. The computation of an AGCD

Polynomial computations can be used for image deblurring because the convolution operation defines the multiplication of two polynomials and the formation of a blurred image by a spatially invariant PSF [12–14,21,22,25,32]. The formation of a blurred image by a spatially invariant PSF in the presence of additive noise is defined in (1), and these polynomial computations require that $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$ and $\mathcal{N}$ be represented as polynomials. It is assumed the PSF is separable, in which case the polynomial representation $H(x, y)$ of $\mathcal{H}$ can be written as

$$H(x,y)=\sum_{k=0}^{p}\sum_{l=0}^{q}h_c(k)x^{p-k}h_r(l)y^{q-l}=H_c(x)H_r(y), \tag{6}$$

where the vertical and horizontal extents of the PSF are $p+1$ and $q+1$ pixels respectively, and

$$H_c(x)=\sum_{k=0}^{p}h_c(k)x^{p-k} \qquad \text{and} \qquad H_r(y)=\sum_{l=0}^{q}h_r(l)y^{q-l},$$

are the polynomial forms of the column and row components of the PSF. It is shown in [32] that the coefficients $h_c(k)$ and $h_r(l)$ can be obtained by computing an AGCD of two arbitrary columns, and an AGCD of two arbitrary rows, respectively, of $\mathcal{G}$, where the pixel values of each row and each column are the coefficients of a polynomial. The deblurred image is then obtained by deconvolving $\mathcal{H}$ from $\mathcal{G}$ [32].

Many methods for the computation of an AGCD of two polynomials have been developed, including methods based on the QR decomposition of the Sylvester matrix [11,37], methods based on the singular value decomposition (SVD) of the Sylvester matrix [10, 15], optimisation methods [9,38] and methods that exploit the structure of the Sylvester matrix [3,4,33,34]. In this paper, the method of SNTLN is applied to the Sylvester matrix in order to compute an AGCD of two inexact (noisy) polynomials.

There are several definitions of an AGCD and they all involve a tolerance ϵ that is a measure of the error in the given inexact polynomials. For example, the following definition of an AGCD of the polynomials $f(y)$ and $g(y)$, termed an ϵ-divisor of $f(y)$ and $g(y)$, is used by Bini and Boito [4].

Definition 3.1. Let $f(y)$ and $g(y)$ be polynomials of degrees m and n respectively. A polynomial $d(y)$ is an ϵ-divisor of $f(y)$ and $g(y)$ if there exist polynomials $\tilde{f}(y)$ and $\tilde{g}(y)$, of degrees m and n respectively, such that

$$\left\|f(y)-\tilde{f}(y)\right\| \leq \epsilon \left\|f(y)\right\| \qquad \text{and} \qquad \left\|g(y)-\tilde{g}(y)\right\| \leq \epsilon \left\|g(y)\right\|, \tag{7}$$

and $d(y)$ divides $\tilde{f}(y)$ and $\tilde{g}(y)$. If $d(y)$ is an ϵ-divisor, of maximum degree, of $f(y)$ and $g(y)$, then it is called an ϵ-GCD (greatest common divisor) of $f(y)$ and $g(y)$. The polynomials $u(y)=\tilde{f}(y)/d(y)$ and $v(y)=\tilde{g}(y)/d(y)$ are called ϵ-cofactors.

This definition is stated in terms of the normwise error ϵ and it is therefore instructive to consider the conditions under which a normwise error in the coefficients of $f(y)$ and $g(y)$ is a good approximation of the componentwise errors. Let these polynomials, and the polynomials $\tilde{f}(y)$ and $\tilde{g}(y)$, be given by

$$f(y)=\sum_{i=0}^{m} a_i y^{m-i} \qquad \text{and} \qquad \tilde{f}(y)=\sum_{i=0}^{m} \tilde{a}_i y^{m-i},$$

and

$$g(y)=\sum_{i=0}^{n} b_i y^{n-i} \qquad \text{and} \qquad \tilde{g}(y)=\sum_{i=0}^{n} \tilde{b}_i y^{n-i},$$

and thus if the componentwise errors λ_i and μ_j in $f(y)$ and $g(y)$ are approximately constant and equal to λ and μ respectively,

$$\left|a_i-\tilde{a}_i\right| \approx \lambda_i \left|a_i\right| \approx \lambda \left|a_i\right| \qquad \text{and} \qquad \left|b_j-\tilde{b}_j\right| \approx \mu_j \left|b_j\right| \approx \mu \left|b_j\right|, \tag{8}$$

for $i=0,\ldots,m$, and $j=0,\ldots,n$, then

$$\left\|f(y)-\tilde{f}(y)\right\| \approx \lambda \left\|f(y)\right\| \qquad \text{and} \qquad \left\|g(y)-\tilde{g}(y)\right\| \approx \mu \left\|g(y)\right\|.$$

If (8) is satisfied and $\lambda \approx \mu$, then the componentwise errors are approximately equal to the normwise error ϵ, which is defined in (7). If, however, (8) is not satisfied, that is, the coefficients a_i, $\tilde{a}_i$, b_j and $\tilde{b}_j$ are such that

$$\frac{\max_{i=0,\ldots,m} \frac{\left|a_i-\tilde{a}_i\right|}{\left|a_i\right|}}{\min_{i=0,\ldots,m} \frac{\left|a_i-\tilde{a}_i\right|}{\left|a_i\right|}} \gg 1 \qquad \text{or} \qquad \frac{\max_{j=0,\ldots,n} \frac{\left|b_j-\tilde{b}_j\right|}{\left|b_j\right|}}{\min_{j=0,\ldots,n} \frac{\left|b_j-\tilde{b}_j\right|}{\left|b_j\right|}} \gg 1,$$

then the normwise error ϵ does not yield information on the componentwise errors λ_i and μ_j.

This discussion has implications for the work considered in this paper because the row and column components of the PSF are computed from AGCD computations, where the polynomials $f(y)$ and $g(y)$ are, respectively, two arbitrary rows and two arbitrary columns, of the blurred image $\mathcal{G}$. The noise in a blurred image obtained in practical problems is not uniformly distributed across the image, and thus a normwise threshold ϵ may yield poor results because it is not a good approximation of the relative error of each pixel in the blurred image. In particular, ϵ may be too small in some regions of the blurred image, in which case not all the blur will be filtered, and it may be too large in other regions of the blurred image, in which case some of the exact image will be filtered. Furthermore, it cannot be assumed that the value of ϵ is known, and even if it is known, it may only be known approximately and not exactly, which makes the specification of a value of ϵ difficult.

Methods for the computation of an AGCD of $f(y)$ and $g(y)$ usually require that ϵ^{-1} be a measure of the SNR. These methods attempt to compute common divisors of degrees $\min(m, n), \min(m, n) - 1, \min(m, n) - 2, \ldots, 2, 1$, and the computations are terminated when (7) is satisfied. The discussion above shows, however, that it is not possible to assign ϵ a meaningful interpretation when deblurring an image because the error varies widely across the image. A method for the computation of an AGCD of two polynomials when this condition is satisfied is discussed in Section 4.

4. The Sylvester matrix for AGCD computations

This section considers the application of the Sylvester matrix to the calculation of an AGCD of two polynomials. This computation is implemented in two stages, where the degree of an AGCD is computed in the first stage and the coefficients of an AGCD of this degree are computed in the second stage. These stages are described in Sections 4.1 and 4.3 respectively, and Section 4.2 considers some properties of the orthogonal matrix Q and upper triangular matrix R from the QR decomposition of the Sylvester matrix because this decomposition is used in Section 4.1.

4.1. The degree of an AGCD

The degree of an AGCD of the inexact polynomials $f(y)$ and $g(y)$ is computed using their Sylvester matrix $S(f, g)$ and its subresultant matrices $S_k(f, g)$, $k = 2, \ldots, \min(m, n)$, where $S_1(f, g) = S(f, g)$ [2]. These polynomials are, by definition, subject to error, but it is appropriate to consider initially the use of the Sylvester matrix for the calculation of the degree of the GCD of the polynomials $\hat{f}(y)$ and $\hat{g}(y)$ that are the exact forms of $f(y)$ and $g(y)$,

$$\hat{f}(y)=\sum_{i=0}^{m}\hat{a}_i y^{m-i} \qquad \text{and} \qquad \hat{g}(y)=\sum_{i=0}^{n}\hat{b}_i y^{n-i}, \tag{9}$$

and then consider the modifications required for the computation of the degree of an AGCD of $f(y)$ and $g(y)$.

The Sylvester matrix $S(\hat{f},\hat{g})\in\mathbb{R}^{(m+n)\times(m+n)}$ is formed by the concatenation of two Tœplitz matrices $G_1(\hat{f})\in\mathbb{R}^{(m+n)\times n}$ and $H_1(\hat{g})\in\mathbb{R}^{(m+n)\times m}$,

$$S(\hat{f},\hat{g})=S_1(\hat{f},\hat{g})=\left[\begin{array}{cc} G_1(\hat{f}) & H_1(\hat{g})\end{array}\right]=\left[\begin{array}{cccccc} \hat{a}_0 & & & \hat{b}_0 & & \\ \hat{a}_1 & \ddots & & \hat{b}_1 & \ddots & \\ \vdots & \ddots & \hat{a}_0 & \vdots & \ddots & \hat{b}_0 \\ \hat{a}_m & \ddots & \hat{a}_1 & \hat{b}_n & \ddots & \hat{b}_1 \\ & \ddots & \vdots & & \ddots & \vdots \\ & & \hat{a}_m & & & \hat{b}_n \end{array}\right].$$

The kth subresultant matrix $S_k(\hat{f},\hat{g})\in\mathbb{R}^{(m+n-k+1)\times(m+n-2k+2)}$,

$$S_k(\hat{f},\hat{g})=\left[\begin{array}{cc} G_k(\hat{f}) & H_k(\hat{g})\end{array}\right], \qquad k=2,\ldots,\min(m,n),$$

where $G_k(\hat{f})\in\mathbb{R}^{(m+n-k+1)\times(n-k+1)}$ and $H_k(\hat{g})\in\mathbb{R}^{(m+n-k+1)\times(m-k+1)}$, is formed by deleting the last $k-1$ columns of $G_1(\hat{f})$, the last $k-1$ columns of $H_1(\hat{g})$, and the last $k-1$ rows of $S_1(\hat{f},\hat{g})$. It follows that the update formula of the QR decomposition allows the QR decomposition of $S_{k+1}(\hat{f},\hat{g})$ to be computed efficiently from the QR decomposition of $S_k(\hat{f},\hat{g})$, and this property is used for the computation of the degree of an AGCD of two inexact polynomials.

The application of $S_k(\hat{f},\hat{g})$ to the calculation of the degree $\hat{t}$ of the GCD $\hat{d}(y)$ of $\hat{f}(y)$ and $\hat{g}(y)$ is considered in Theorem 4.1.

Theorem 4.1. *Let $\hat{f}(y)$ and $\hat{g}(y)$ be defined in (9). The value of $\hat{t}=\deg\hat{d}(y)$ is equal to the largest integer k such that $S_k(\hat{f},\hat{g})$ is singular,*

$$\begin{aligned} &\textit{rank } S_k(\hat{f},\hat{g})<m+n-2k+2, \quad k=1,\ldots,\hat{t},\\ &\textit{rank } S_k(\hat{f},\hat{g})=m+n-2k+2, \quad k=\hat{t}+1,\ldots,\min(m,n), \end{aligned}$$

and $S(\hat{f},\hat{g})$ is non-singular if and only if $\hat{f}(y)$ and $\hat{g}(y)$ are coprime.

The application of Theorem 4.1 to the inexact polynomials $f(y)$ and $g(y)$ requires that they be preprocessed by three operations [33–35], where the second and third operations introduce the parameters $\alpha_0>0$ and $\theta_0>0$ respectively, such that $y=\theta_0 w$ where w is the new independent variable, that is, an AGCD of the polynomials $\bar{f}(w)=f(\theta_0 w)$ and $\alpha_0\bar{g}(w)=\alpha_0 g(\theta_0 w)$ is computed. The values α_0 and θ_0 are refined in the iterative procedure for the calculation of an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$, and if θ^* is the value of θ_0 at the termination of this procedure, then the inverse transformation $w = y/\theta^*$ is applied to the results of the AGCD computation in order that the results be expressed in the same independent variable, y, as the specified polynomials $f(y)$ and $g(y)$.

Theorem 4.1 shows that the computation of the degree t of an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ requires the calculation of the rank of their Sylvester matrix $S(\bar{f}, \alpha_0\bar{g})$ and each subresultant matrix $S_k(\bar{f}, \alpha_0\bar{g})$. Computational experiments in [35] show that the SVD of $S(\bar{f}, \alpha_0\bar{g})$ does not yield a good estimate of t, but two other methods, which yield good results and are based on Theorem 4.1, are described in [31,35,36]. They are, however, expensive because they require the computation of the SVD of $S_k(\bar{f}, \alpha_0\bar{g})$ for each value of $k = 1, \ldots, \min(m, n)$, and the update formula for the SVD is more complicated than its equivalent for the QR decomposition. It is therefore better to determine the value of t from the QR decomposition of each matrix $S_k(\bar{f}, \alpha_0\bar{g})$, as noted above, and this issue is now considered.

Let Q_kR_k be the QR decomposition of $S_k(\bar{f}, \alpha_0\bar{g})$, where the diagonal entries of R_k are $R_{k,i,i}, i = 1, \ldots, m+n-2k+2$. The singularity, or otherwise, of $S_k(\bar{f}, \alpha_0\bar{g})$ manifests itself in the value of μ_k because if

$$\mu_k = \frac{\max_i |R_{k,i,i}|}{\min_i |R_{k,i,i}|}, \qquad k = 1, \ldots, \min(m, n), \tag{10}$$

then the conditions $\mu_k \gg 1$ and $\mu_k \approx 1$ imply $S_k(\bar{f}, \alpha_0\bar{g})$ is, respectively, numerically singular and of full rank. It follows that the degree t of an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ can be calculated from the values of μ_k, and the criterion for its determination is derived by considering the values of μ_k for the exact polynomials $\hat{f}(y)$ and $\hat{g}(y)$, whose GCD is of degree $\hat{t}$. In this circumstance, it follows from Theorem 4.1 and (10) that the values of μ_k satisfy

$$\mu_k = \begin{cases} \infty, & k = 1, \ldots, \hat{t}, \\ \gamma_k < \infty, & k = \hat{t}+1, \ldots, \min(m, n), \end{cases}$$

because at least one diagonal entry of R_k is zero for $k = 1, \ldots, \hat{t}$. The value of $\hat{t}$ is determined by the change from an infinite value of μ_k to a finite value of μ_k, and the extension of this criterion from the exact polynomials $\hat{f}(y)$ and $\hat{g}(y)$ to the inexact polynomials $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ is therefore given by

$$t = \arg\min_k \left(\mu_k - \mu_{k+1}\right), \qquad k = 1, \ldots, \min(m, n) - 1. \tag{11}$$

An incorrect value of t may be computed because the values of μ_k are subject to error, in which case the computed PSF will have a large error. This error can be minimised by recalling that the horizontal and vertical components of the PSF are computed by choosing two arbitrary rows and two arbitrary columns, respectively, of the blurred

Algorithm 1 The calculation of the degree of the horizontal or vertical component of the polynomial form of the PSF.

```
Input   A blurred image $\mathcal{G}$.
Output  The degree $t$ of the horizontal or vertical component of the polynomial form of the PSF.
Begin
        % Calculate the size of $\mathcal{G}$ and define the number of pairs of rows or columns
        % of $\mathcal{G}$ used for the calculation of $t$.
        [MG,NG] = size($\mathcal{G}$)
        trials = 25   % the number of trials for the calculation of $t$
        % Initialise the array that stores the results. The size of the array depends
        % on whether the row or column component of the PSF is considered.
        if row component
          upperlimitk = NG-1
        else
          upperlimitk = MG-1
        end if
        results = zero(upperlimitk,1)
        for s=1:1:trials   % loop for the number of pairs of rows or columns
          1. Select two rows or two columns from $\mathcal{G}$ and denote their polynomial forms $f(y)$ and $g(y)$.
          2. Preprocess $f(y)$ and $g(y)$ to yield the polynomials $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$.
          3. Calculate $Q_1R_1$, the QR decomposition of $S_1(\bar{f},\alpha_0\bar{g}) = S(\bar{f},\alpha_0\bar{g})$.
          4. Calculate $\mu_1$ from (10).
          % Calculate $\mu_k$ for the subresultant matrices.
          5. for k=2:1:upperlimitk
                 Form $S_k(\bar{f},\alpha_0\bar{g})$ from $S_{k-1}(\bar{f},\alpha_0\bar{g})$.
                 Calculate the QR decomposition $Q_kR_k$ of $S_k(\bar{f},\alpha_0\bar{g})$ from the QR decomposition
                 $Q_{k-1}R_{k-1}$ of $S_{k-1}(\bar{f},\alpha_0\bar{g})$.
                 Calculate the value of $\mu_k$ from (10).
             end
          % Calculate the value of $t$ from (11) for the trial.
          results(t) = results(t)+1
        end
        % The computed value of $t$ is equal to the mode of the entries in the vector results.
          t = mode(results)
End
```

image $\mathcal{G}$. If the computation of the degrees of the components of the PSF is performed several times, for different pairs of rows and different pairs of columns, then the modes of the results enable the values of t for these components to be computed reliably. This calculation for t is described in Algorithm 1.

It is shown by Stewart [30] that an error of order ϵ in an arbitrary matrix $A = QR$ can induce an error of order $\epsilon\kappa(A)$ in Q and R, where $\kappa(A)$ is the condition number of A. Chang and Paige [8] show, however, that the bound $\epsilon\kappa(A)$ is a large overestimate for most problems, and they develop a bound that more accurately reflects the condition of R. Example 4.1 confirms the improved bound in [8].

Example 4.1. Fig. 1(a) shows the PSF to which uncertainty $\mathcal{E}$ was added, and this perturbed PSF was convolved with the image shown in Fig. 1(b). Noise $\mathcal{N}$ was then added, and thus the blurring model is an extension of (1),

$$\mathcal{G} = (\mathcal{H} + \mathcal{E}) * \mathcal{F} + \mathcal{N}. \tag{12}$$

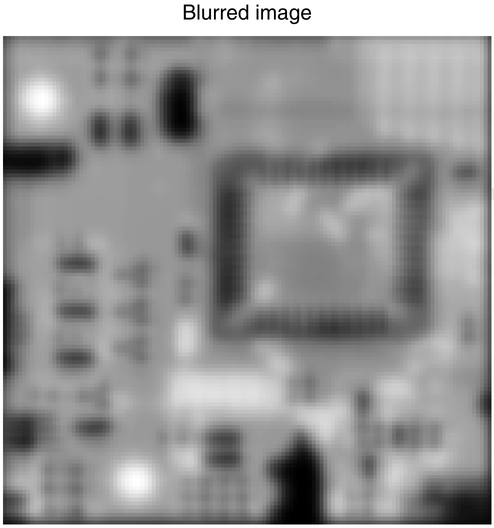


Fig. 5. The blurred image for Example 4.1.

If the coefficients of the polynomial forms $E(x,y)$, $H(x,y)$ and $N(x,y)$ of $\mathcal{E}$, $\mathcal{H}$ and $\mathcal{N}$ are $e(i,j)$, $h(i,j)$ and $n(i,j)$ respectively, and the coefficients of

$$S(x,y) = \Big(H(x,y) + E(x,y)\Big)F(x,y),$$

are $s(i,j)$, then the uncertainty $e(i,j)$ and error $n(i,j)$ were chosen to satisfy

$$0 < \frac{e(i,j)}{h(i,j)} \leq r_{i,j}, \qquad i = 0,\dots,p, \qquad j = 0,\dots,q, \tag{13}$$

and

$$0 < \frac{n(i,j)}{s(i,j)} \leq s_{i,j}, \quad i = 0,\dots,M+p-1, \quad j = 0,\dots,N+q-1, \tag{14}$$

where $r_{i,j}$ and $s_{i,j}$ are uniformly distributed random variables in the range $\big[10^{-5}, 10^{-4}\big]$. Fig. 5 shows the blurred image that results from these perturbations.

Definition 3.1 of an AGCD cannot be applied because the upper bounds of the relative errors of the PSF and additive noise are not constant, and it is therefore difficult to set a threshold for a termination criterion in an iterative procedure for the calculation of the degree of an AGCD.

Figs. 6 and 7 show the variation of $\log_{10}\mu_k$ with k for the calculation of the degree of the row component of the PSF, where the graphs are obtained with different pairs of randomly selected rows of $\mathcal{G}$. Fig. 6 yields the correct result because the value of t, computed from (11), is equal to 8. Fig. 7 yields, however, $t = 4$, which is incorrect, although the graph in the figure is identical in shape to the graph in Fig. 6. Since

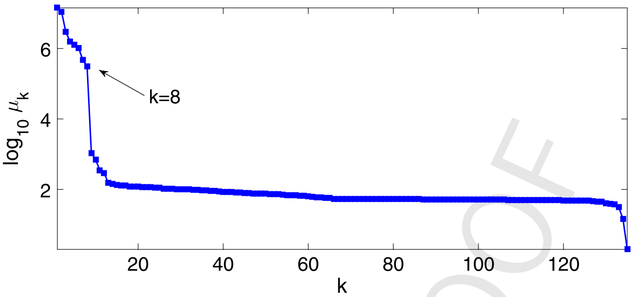


Fig. 6. The ratio $\log_{10} \mu_k$ against the order k of the subresultant matrix. The correct result, $k = t = 8$, for the degree of the row component of the PSF is obtained.

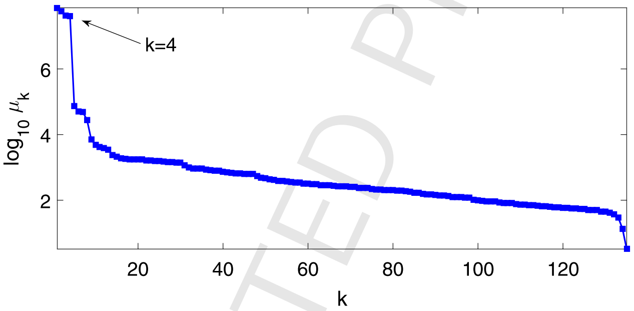


Fig. 7. The ratio $\log_{10} \mu_k$ against the order k of the subresultant matrix. An incorrect result, $k = t = 4$, for the degree of the row component of the PSF is obtained.

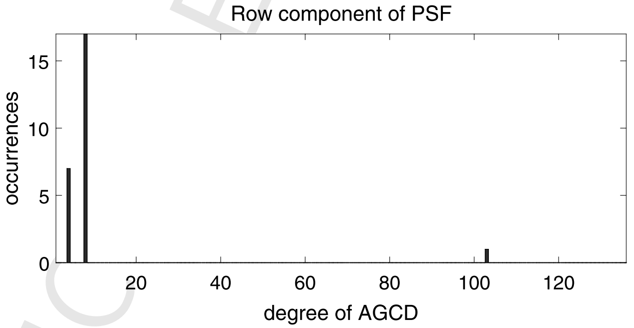


Fig. 8. The histogram of the results for the computation of the degree of the row component of the PSF for the image in Fig. 5.

there does not exist a feature that enables the results in Figs. 6 and 7 to be classified as, respectively, correct and incorrect, the experiment was performed 25 times, using randomly selected pairs of rows of $\mathcal{G}$. The histogram of the results is shown in Fig. 8, and Fig. 9 shows the histogram for the degree of the column component of the PSF. The correct results for the degrees of the row and column components of the PSF, $t = 8$ and $t = 10$ respectively, are obtained in 17 of the 25 experiments, and 23 of the 25

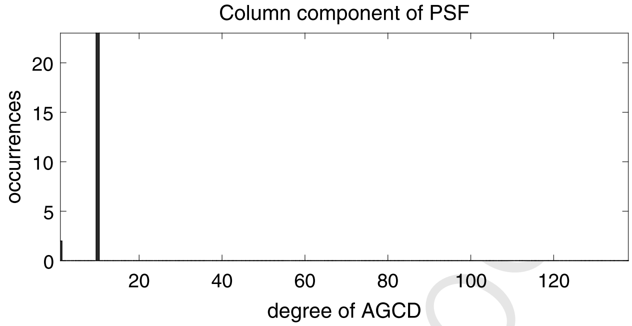


Fig. 9. The histogram of the results for the computation of the degree of the column component of the PSF for the image in Fig. 5.

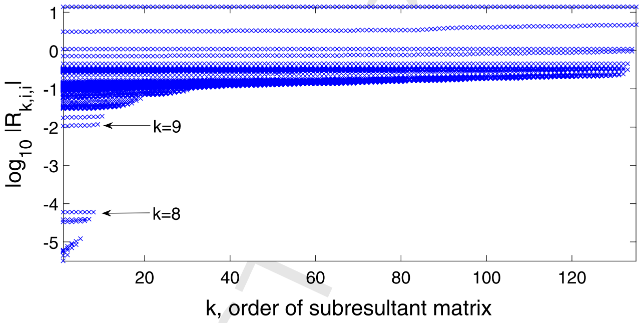


Fig. 10. The diagonal entries $R_{k,i,i}$ of R_k of the kth subresultant matrix for the situation shown in Fig. 6.

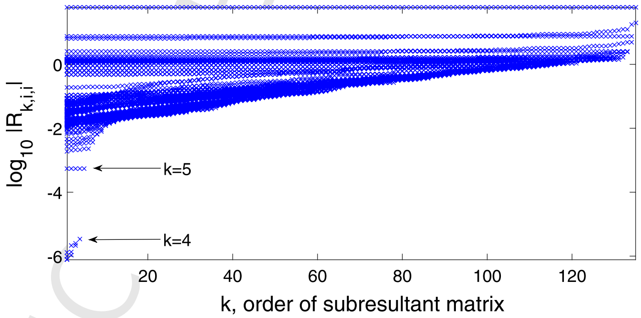


Fig. 11. The diagonal entries $R_{k,i,i}$ of R_k of the kth subresultant matrix for the situation shown in Fig. 7.

experiments. It follows that the correct degrees of the row and column components of the PSF are obtained when all the results from the 25 trials are considered.

Figs. 6, 7, 8 and 9 show that the diagonal entries of R_k are important for the calculation of the degrees of the row and column components of the PSF. Figs. 10 and 11 show the diagonal entries $R_{k,i,i}$ of R_k for each value of the order k of the subresultant matrix. They correspond to, respectively, the situations in which the QR decomposition yields the correct value and an incorrect value of the degree of the row component of the PSF, and they are therefore associated with Figs. 6 and 7 respectively. Figs. 10 and 11 are very similar in form, and they therefore confirm that the computation of the horizontal and vertical extents of the PSF from one pair of rows, and one pair of columns, may yield incorrect results. □

4.2. Properties of the QR decomposition of $S_k(\bar{f}, \alpha_0\bar{g})$

It was shown in Section 4.1 that the computation of the degree of an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ uses the QR decomposition of their Sylvester matrix and subresultant matrices $S_k(\bar{f}, \alpha_0\bar{g})$, $k = 1, \ldots, \min(m, n)$. The structured nature of these matrices manifests itself in properties of Q_k and R_k, and this section considers some of these properties. Since the first $n - k + 1$ columns and last $m - k + 1$ columns of $S_k(\bar{f}, \alpha_0\bar{g})$ contain the coefficients of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ respectively, and these polynomials are, in general, independent, some results must be stated separately for these two groups of columns.

Equation (10) requires that the diagonal entry of maximum absolute value of each matrix R_k, $k = 1, \ldots, \min(m, n)$, be calculated, and Theorem 4.2 shows that if $\left\|\bar{\mathbf{f}}\right\|_2 \geq \alpha_0 \left\|\bar{\mathbf{g}}\right\|_2$, then these values are constant and equal to $\left\|\bar{\mathbf{f}}\right\|_2$, where $\mathbf{p}$ is the vector of the coefficients of the polynomial $p(y)$.

Theorem 4.2. *Let Q_kR_k be the QR decomposition of $S_k(\bar{f}, \alpha_0\bar{g})$ and let $R_{k,i,j}$ be element (i, j) of R_k. If $\left\|\bar{\mathbf{f}}\right\|_2 \geq \alpha_0 \left\|\bar{\mathbf{g}}\right\|_2$, then*

$$\max |R_{k,i,j}| = |R_{k,1,1}| = \left\|\bar{\mathbf{f}}\right\|_2, \tag{15}$$

for $i = 1, \ldots, m + n - 2k + 2$, $j = i, \ldots, m + n - 2k + 2$, and $k = 1, \ldots, \min(m, n)$.

Proof. Consider the ith column $\mathbf{S}_{k,i} = S_k\mathbf{e}_i = Q_kR_k\mathbf{e}_i$ of S_k, where $\mathbf{e}_i$ is the ith unit basis vector. It follows that $\mathbf{R}_{k,i} = R_k\mathbf{e}_i$ is the ith column of R_k, and thus

$$\left\|\mathbf{S}_{k,i}\right\|_2 = \left\|\mathbf{R}_{k,i}\right\|_2 = \begin{cases} \left\|\bar{\mathbf{f}}\right\|_2, & i = 1, \ldots, n - k + 1, \\ \alpha_0 \left\|\bar{\mathbf{g}}\right\|_2, & i = n - k + 2, \ldots, m + n - 2k + 2, \end{cases} \tag{16}$$

for $k = 1, \ldots, \min(m, n)$. Consider this equation for the first column, $i = 1$, of R_k,

$$\left\|\mathbf{R}_{k,1}\right\|_2 = |R_{k,1,1}| = \left\|\bar{\mathbf{f}}\right\|_2, \qquad k = 1, \ldots, \min(m, n), \tag{17}$$

that is, the absolute value of entry $(1, 1)$ of R_k is equal to $\left\|\bar{\mathbf{f}}\right\|_2$ for all $k = 1, \ldots, \min(m, n)$. Equation (15) then follows from (16) and (17). □

Theorem 4.3 shows that the first $n - k + 1$ columns of R_k are related by an orthogonal matrix.

Theorem 4.3. *The $(i+j)$th column $\mathbf{R}_{k,i+j}$ of R_k is formed by a rotation of the ith column $\mathbf{R}_{k,i}$ of R_k, where $i = 1, \ldots, n-k+1$, and $j = 0, \ldots, n-k+1-i$.*

Proof. Let H be a square matrix of order $m+n-k+1$, such that if $\mathbf{c}_i \in \mathbb{R}^{m+n-k+1}$ is the ith column of $S_k(\bar{f}, \alpha_0\bar{g})$, then the lower triangular Tœplitz structure of the first $n-k+1$ columns of $S_k(\bar{f}, \alpha_0\bar{g})$ implies

$$\mathbf{c}_{i+1} = H\mathbf{c}_i, \qquad H = \begin{bmatrix} \mathbf{0}_{1,m+n-k} & 1 \\ I_{m+n-k} & \mathbf{0}_{m+n-k,1} \end{bmatrix},$$

where the subscripts on the zero matrices indicate their order, H is orthogonal,

$$\mathbf{c}_{i+j} = H^j\mathbf{c}_i, \qquad \mathbf{e}_{i+j} = H^j\mathbf{e}_i, \tag{18}$$

$\mathbf{e}_i \in \mathbb{R}^{m+n-k+1}$, $i = 1, \ldots, n-k+1$, and $j = 0, \ldots, n-k+1-i$. It follows from the definition of $\mathbf{c}_i$ that $\mathbf{c}_i = Q_kR_k\mathbf{e}_i$, and thus (18) yields

$$Q_kR_k\mathbf{e}_{i+j} = H^jQ_kR_k\mathbf{e}_i,$$

and since $\mathbf{R}_{k,i}$ is the ith column of R_k,

$$\mathbf{R}_{k,i+j} = Q_k^TH^jQ_k\mathbf{R}_{k,i}, \tag{19}$$

where $Q_k^TH^jQ_k$ is an orthogonal matrix. It therefore follows that the $(i+j)$th column of R_k is formed by a rotation of the ith column of R_k. □

Equation (19) is valid for the first $n-k+1$ columns of R_k only, and an equation of the same form, with different limits on the subscripts, applies to the last $m-k+1$ columns of R_k,

$$\mathbf{R}_{k,i+j} = Q_k^TH^jQ_k\mathbf{R}_{k,i}, \tag{20}$$

for $i = n-k+2, \ldots, m+n-2k+2$, and $j = 0, \ldots, m+n-2k+2-i$. Equations (19) and (20) are refined forms of (16) because they show explicitly the relationship between the columns of R_k, rather than merely the relationship between the norms of the vectors formed from these columns.

Interest in this work is focused on the diagonal entries $R_{k,i,i}$ of R_k, and it follows from (18) and (19) that for $i = 1, \ldots, n-k+1$, and $j = 0, \ldots, n-k+1-i$,

$$R_{k,i+j,i+j} = \mathbf{e}_{i+j}^T\mathbf{R}_{k,i+j} = \mathbf{e}_{i+j}^TQ_k^TH^jQ_k\mathbf{R}_{k,i} = \mathbf{e}_i^T\left(Q_kH^j\right)^TH^jQ_k\mathbf{R}_{k,i},$$

which shows that entry $(i+j, i+j)$ of R_k is a weighted combination of the non-zero entries in the ith column $\mathbf{R}_{k,i}$ of R_k, where the weights are equal to the first i entries in the ith row of the orthogonal matrix $\left(Q_kH^j\right)^TH^jQ_k$.

4.3. The coefficients of an AGCD

The computation of the degree t of an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ was considered in Section 4.1, and this section describes the method of SNTLN for the computation of the coefficients of an AGCD of degree t. This computation must be performed twice, once for the row component, and once for the column component, of the PSF, and the selection of the pair of rows and the pair of columns of the blurred image $\mathcal{G}$ to be used for this computation must be considered. Specifically, it was shown in Section 4.1 that the computation of the degrees of the row and column components of the PSF is performed 25 times, and as shown in Figs. 8 and 9, the degree of the AGCD is equal to the mode of the distribution of t for each component of the PSF. The indices of the pairs of rows and the pairs of columns are recorded for each of the 25 experiments, and the computed value of t is also recorded for each of these pairs. The coefficients of the row component of the PSF are computed from a randomly selected pair of rows that returned the mode of the distribution of t for this component of the PSF, and the same procedure is used for the computation of the coefficients of the column component of the PSF.

It is shown in [32–34] that the computation of the coefficients of an AGCD, as described above, requires the solution a non-linear constrained minimisation problem. The linearisation of this problem leads to a least squares minimisation with an equality constraint, the LSE problem, at each iteration,

$$\min_{\delta\mathbf{y}^{(j)}} \left\| \delta\mathbf{y}^{(j)} - \mathbf{h}^{(j)} \right\|_2 \quad \text{subject to} \quad E^{(j)}\delta\mathbf{y}^{(j)} = \mathbf{t}^{(j)}, \qquad j = 1, 2, \ldots, \tag{21}$$

where $\mathbf{y}^{(j)} = \mathbf{y}^{(j-1)} + \delta\mathbf{y}^{(j)}$, and $\mathbf{y}^{(j)}$ contains the coefficients of the coprime polynomials of the AGCD computation at the jth iteration. The matrices $E^{(j)}$ in (21) are derived from the Sylvester matrix of the polynomial representations of two rows or two columns of the blurred image, as stated above. The LSE problem (21) is solved by the QR decomposition [16], but it can also be solved by the penalty method. The QR decomposition is preferred because it does not require a parameter $\tau \gg 1$, the use of which may cause numerical problems because although the constraint is imposed more severely as τ increases, the coefficient matrix of the linear algebraic equation becomes more ill-conditioned [1,23]. The penalty method is used in [24,26] for the solution of a constrained minimisation problem that arises when regularisation is used for the solution of the BID problem. These numerical issues are not addressed in [24], and the procedure adopted in [26] requires the computation of deblurred images for several values of τ and the selection of the image for which the relative error is a minimum. The authors note that the deblurred image may be sensitive to the value of τ, which confirms the potential ill-conditioning that may arise, as noted above.

The discussion above shows that the same computation is used for the coefficients of the row and column components of the PSF because of the separable form of the PSF (6). The PSF is then deconvolved from $\mathcal{G}$ by solving two sets of least squares problems, thereby forming a deblurred image [32].

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010

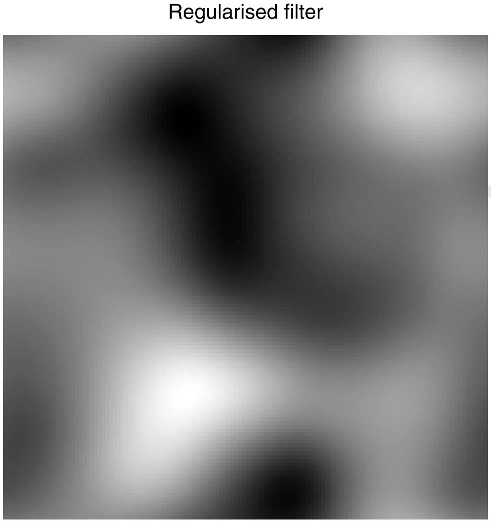


Fig. 12. The deblurred image obtained from `deconvreg.m` with $\texttt{NP} = 1.70 \times 10^7$ and $\lambda = 10^{-9}$.

5. Results

This section compares the deblurred images obtained from the method considered in this paper and the four functions in Matlab discussed in Section 2. The exact and blurred images are shown in Figs. 1(b) and 5 respectively, where the blurred image is formed by the addition of random noise $\mathcal{N}$, and uncertainty $\mathcal{E}$ to the PSF $\mathcal{H}$, as shown in (12), and (13) and (14) are satisfied.

The noise power and PNSR of the blurred image are 1.70×10^7 and 4.86×10^{-2} respectively (4), and Figs. 12 and 13 show the deblurred images from `deconvreg.m` and `deconvwnr.m` obtained with these parameters, after the function `edgetaper.m` was called. The reason for the poor result obtained from `deconvreg.m` can be seen from Fig. 3(b), which shows that the value $\texttt{NP} = 1.70 \times 10^7$ yields a large error in the deblurred image. Similarly, Fig. 4 shows that the value $\texttt{NSR} = 4.86 \times 10^{-2}$ for `deconvwnr.m` is too large.

Fig. 2 shows the variation of the errors in the deblurred image and computed PSF with the number of iterations and the componentwise relative error ε for the function `deconvblind.m`. It is seen that the error in the deblurred image is large and the error in the PSF is small if the number of iterations is small, and that the error in the deblurred image decreases as the number of iterations increases. Fig. 14 shows the deblurred image from `deconvblind.m` with $\texttt{NUMIT} = 100$, which achieves a compromise between the fidelity of the deblurred image and the computed PSF.

Fig. 3(a) shows that the value $\texttt{NUMIT} = 200$ for `deconvlucy.m` yields a deblurred image with a small error, and Fig. 15 shows the deblurred image from `deconvlucy.m` with this parameter value. Figs. 3(b) and 4 show that the optimal parameter values for the functions `deconvreg.m` and `deconwnr.m` are $\texttt{NP} = 1000$ and $\texttt{NSR} = 10^{-4.5} = 3.16 \times 10^{-5}$ respectively, and Figs. 16 and 17 show the deblurred images obtained from

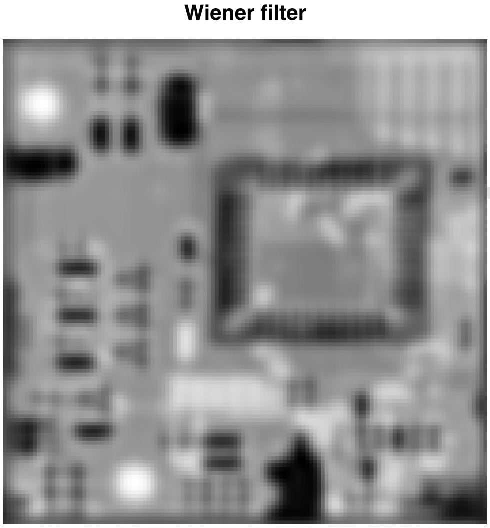


Fig. 13. The deblurred image obtained from `deconvwnr.m` with `NSR` $= 4.86 \times 10^{-2}$.

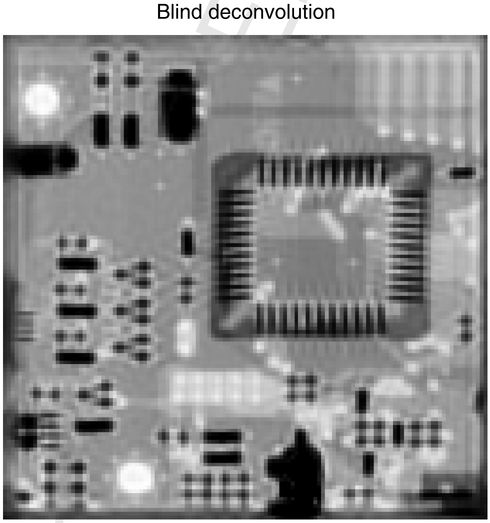


Fig. 14. The deblurred image obtained from `deconvblind.m` with `NUMIT` $= 100$.

these functions. The improvement with respect to the images in Figs. 12 and 13 is clear, particularly for the function `deconvreg.m`. It follows that the selection of the correct parameter values is important, and that the true values of, for example, the noise power and the PNSR, may not yield the best deblurred image [17].

Fig. 18 shows the deblurred image obtained using the method discussed in this paper, and its improvement with respect to the deblurred images obtained by the four functions in Matlab is quantified in Table 1, which shows the relative errors in the deblurred images in Figs. 12–18. The method discussed in this paper yields a deblurred image

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010

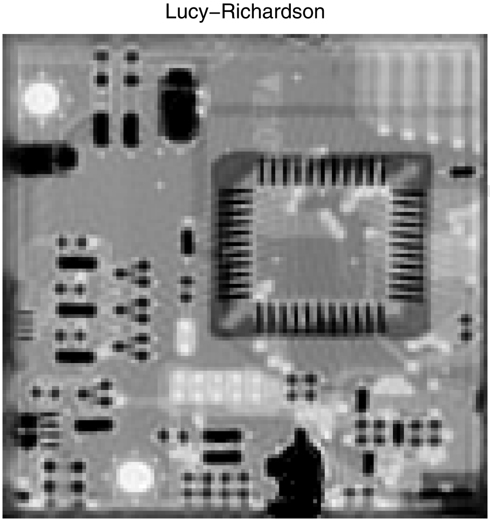


Fig. 15. The deblurred image obtained from `deconvlucy.m` with `NUMIT` = 200.

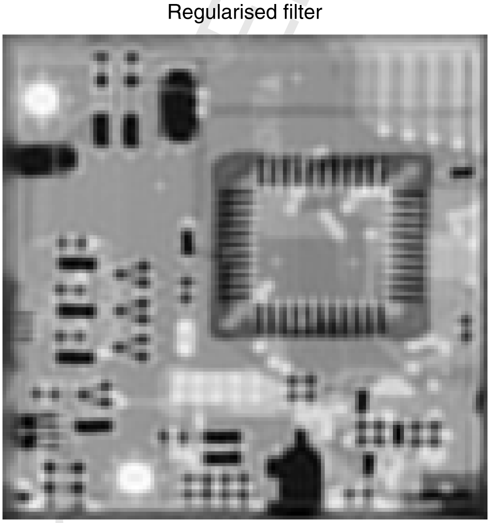


Fig. 16. The deblurred image obtained from `deconvreg.m` with `NP` = 1000 and $\lambda = 10^{-9}$.

whose relative error is about two orders of magnitude smaller than the relative errors in the deblurred images produced by the other methods, even though it is totally blind, and, as explained in Section 2, the PSF was specified in the calls to the four functions in MATLAB.

The deblurred images in Figs. 14–17 still have dark edges, even though the function `edgetaper.m` was used before the four functions in MATLAB were called. These dark edges are not present in the original image, which is shown in Fig. 1(b), and the deblurred image in Fig. 18, which is obtained using the method discussed in this paper. These dark

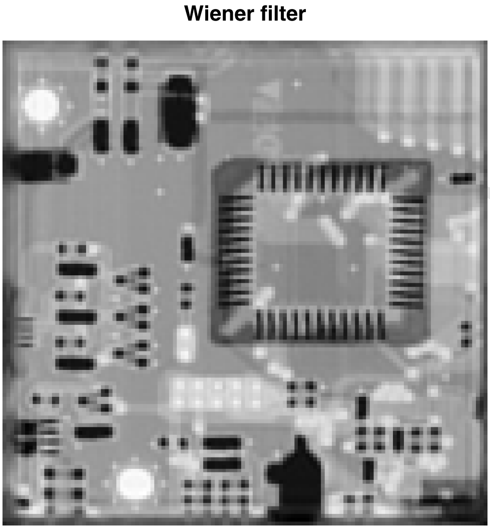


Fig. 17. The deblurred image obtained from `deconvwnr.m` with `NSR` $= 3.16 \times 10^{-5}$.

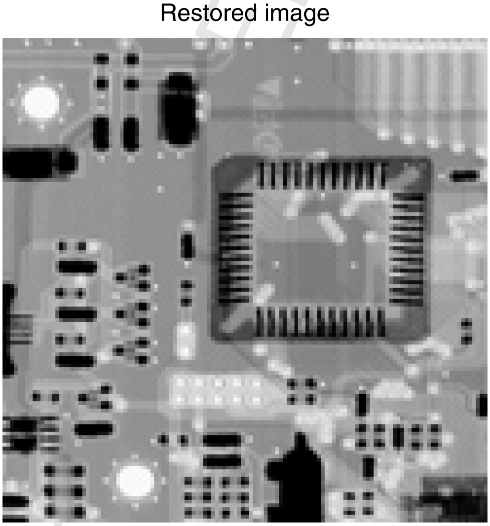


Fig. 18. The deblurred image obtained from the method of image deblurring discussed in this paper.

edges contribute to the errors in the deblurred images, which are equal to, from Table 1, about 0.13 for the deblurred images obtained from the functions in MATLAB when the optimal parameter values are used. This error must be compared with the relative error of the blurred image, which is equal to, from (4), $\left(4.86 \times 10^{-2}\right)^{\frac{1}{2}} = 0.22$.

The AGCD computations and polynomial deconvolutions used in this paper are implemented by the method of SNTLN, and its effectiveness for the numerically robust implementation of these operations is shown in [31,36], where it is used for the computation of multiple roots of a polynomial. The results in Table 1 also show the effectiveness

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010

Table 1
The results of four deblurring methods applied to the image in Fig. 5. The method discussed in this paper is denoted 'Polynomial'.

Method	PSF	Relative error of deblurred image
Regularised filter, Fig. 12	PSF specified	3.40 exp −01
Wiener filter, Fig. 13	PSF specified	1.89 exp −01
Blind deconvolution, Fig. 14	PSF specified	1.32 exp −01
Lucy–Richardson, Fig. 15	PSF specified	1.30 exp −01
Regularised filter, Fig. 16	PSF specified	1.38 exp −01
Wiener filter, Fig. 17	PSF specified	1.24 exp −01
Polynomial, Fig. 18	PSF not specified	5.55 exp −03

of this method for polynomial computations, and in particular, computations on the Sylvester matrix. This matrix is linear and it may therefore be thought that the method of structured total least norm (STLN) [27], which is the simplification of the method of SNTLN to the preservation of a linear structure in a matrix, is adequate. The following discussion shows, however, that it is advantageous to use the method of SNTLN rather than the method of STLN.

It was stated briefly in Section 4.1 that the polynomials $f(y)$ and $g(y)$ must be processed before computations are performed on their Sylvester matrix and subresultant matrices $S_k(f,g)$, $k = 1,\dots,\min(m,n)$, in order to avoid computational problems that may arise [33–35]. Two of these operations introduce the parameters α_0 and θ_0, whose initial values are computed from the solution of a linear programming problem. Their introduction implies that the Sylvester matrix has a non-linear structure for which the method of SNTLN must be used, and they can be considered as degrees of freedom that can be optimised to obtain improved results. They are refined in the iteration (21), and the examples in [35] compare the results from the method of STLN, which is equivalent to the specification $\alpha_0 = \theta_0 = 1$ for all values of j in (21), and the method of SNTLN. These examples show that the method of STLN may return an incorrect value for the degree of an AGCD, that is, the horizontal and vertical extents of the PSF may be in error, and that significantly better results are obtained when the method of SNTLN is used. The work described in this paper differs, therefore, from the work in [24,26] because the method of STLN is used in these papers and, as discussed in Section 4.3, a constrained minimisation problem is solved by the penalty method.

The normwise relative error in the computed PSF using the method discussed in this paper is 1.08×10^{-4}, which is approximately equal to the upper bound of the componentwise relative error in the PSF (13).

6. Summary

This paper has considered the application of polynomial computations for the removal of blur from an image. It has been shown that the degrees of the row and column components of the PSF can be computed efficiently by the QR decomposition of the Sylvester matrix and its subresultant matrices. The coefficients of the PSF are computed from two applications, one for the row component of the PSF and one for the column component of the PSF, of the method of SNTLN to an approximate linear algebraic equation whose coefficient matrix and right hand side vector are derived from a Sylvester subresultant matrix. The deblurred image is then obtained from two sets of deconvolutions, which are solved in the least squares sense.

A parameter study of the four functions in MATLAB that perform BID was performed and it was shown that computational experiments are required to determine the parameters that yield the best deblurred image for each function. These deblurred images were then compared with the deblurred image obtained from the method discussed in this paper, and it was shown that the relative error of this deblurred image is much smaller than the relative errors of the deblurred images obtained from the functions in MATLAB.

References

[1] A. Anda, H. Park, Self-scaling fast rotations for stiff and equality-constrained linear least squares problems, Linear Algebra Appl. 234 (1996) 137–161.

[2] S. Barnett, Polynomials and Linear Control Systems, Marcel Dekker, New York, USA, 1983.

[3] D.A. Bini, P. Boito, Structured matrix-based methods for polynomial ϵ-GCD: analysis and comparisons, in: Proc. Int. Symp. Symbolic and Algebraic Computation, Waterloo, Canada, 2007, pp. 9–16.

[4] D.A. Bini, P. Boito, A fast algorithm for approximate polynomial GCD based on structured matrix computations, in: D.A. Bini, V. Mehrmann, V. Olshevsky, E. Tyrtshnikov, M. Van Barel (Eds.), Numerical Methods for Structured Matrices and Applications: The Georg Heinig Memorial Volume, Birkhäuser, 2010, pp. 155–173.

[5] A. Bouhamidi, K. Jbilou, A Kronecker approximation with a convex constrained optimization method for blind image restoration, Optim. Lett. 6 (2012) 1251–1264.

[6] A.S. Carasso, Linear and nonlinear image deblurring: a documented study, SIAM J. Numer. Anal. 36 (6) (1999) 1659–1689.

[7] A.S. Carasso, Direct blind deconvolution, SIAM J. Appl. Math. 61 (6) (2001) 1980–2007.

[8] X. Chang, C. Paige, A perturbation analysis for R in the QR factorization, Technical report, No. SOCS-95.7, School of Computer Science, McGill University, Montreal, Canada, 1995.

[9] P. Chin, R.M. Corless, Optimization strategies for the approximate GCD problem, in: Proc. Int. Symp. Symbolic and Algebraic Computation, Rostock, Germany, 1998, pp. 228–235.

[10] R.M. Corless, P.M. Gianni, B.M. Trager, S.M. Watt, The singular value decomposition for polynomial systems, in: Proc. Int. Symp. Symbolic and Algebraic Computation, ACM Press, New York, 1995, pp. 195–207.

[11] R.M. Corless, S.M. Watt, L. Zhi, QR factoring to compute the GCD of univariate approximate polynomials, IEEE Trans. Signal Process. 52 (12) (2004) 3394–3402.

[12] A. Danelakis, M. Mitrouli, D. Triantafyllou, Blind image deconvolution using a banded matrix method, Numer. Algorithms 64 (2013) 43–72.

[13] G. Diaz-Toca, S. Belhaj, Blind image deconvolution through Bezoutians, in: 8th International Congress on Industrial and Applied Mathematics, Beijing, China, 2015.

[14] S. El-Khamy, M. Hadhoud, M. Dessouky, B. Salam, F. El-Samie, A greatest common divisor approach to blind super-resolution reconstruction of images, J. Modern Opt. 53 (8) (2006) 1027–1039.

[15] I. Emiris, A. Galligo, H. Lombardi, Certified approximate univariate GCDs, J. Pure Appl. Algebra 117 (1997) 118.

[16] G.H. Golub, C.F. Van Loan, Matrix Computations, John Hopkins University Press, Baltimore, USA, 2013.

[17] R.C. Gonzalez, R.E. Woods, S.L. Eddins, Digital Image Processing Using MATLAB, Gatesmark Publishing, 2009.

[18] B. Gunturk, X. Li, Image Registration: Fundamentals and Advanves, CRC Press, Florida, USA, 2013.

[19] D. Kundur, D. Hatzinakos, Blind image deconvolution, IEEE Signal Process. Mag. 13 (3) (1996) 43–64.

Please cite this article in press as: J.R. Winkler, Polynomial computations for blind image deconvolution, Linear Algebra Appl. (2015), http://dx.doi.org/10.1016/j.laa.2015.10.010

[20] R.G. Lane, R.H.T. Bates, Auotmatic multidimensional deconvolution, J. Opt. Soc. Amer. 4 (1) (1987) 180–188.
[21] Z. Li, Z. Yang, L. Zhi, Blind image deconvolution via fast approximate GCD, in: Proc. Int. Symp. Symbolic and Algebraic Computation, 2010, pp. 155–162.
[22] B. Liang, S. Pillai, Blind image deconvolution using a robust 2-D GCD approach, IEEE Int. Symp. Circuits Syst. (1997) 1185–1188.
[23] C. Van Loan, On the method of weighting for equality-constrained least squares problems, SIAM J. Numer. Anal. 22 (5) (1985) 851–864.
[24] S. Oh, S. Kwon, J. Heon Yun, A method for structured linear total least norm on blind deconvolution problem, J. Appl. Math. Comput. 19 (2005) 151–164.
[25] S. Pillai, B. Liang, Blind image deconvolution using a robust GCD approach, IEEE Trans. Image Process. 8 (2) (1999) 295–301.
[26] A. Pruessner, D.P. O'Leary, Blind deconvolution using a regularized structured total least norm algorithm, SIAM J. Matrix Anal. Appl. 24 (4) (2003) 1018–1037.
[27] J. Ben Rosen, H. Park, J. Glick, Total least norm formulation and solution for structured problems, SIAM J. Matrix Anal. Appl. 17 (1) (1996) 110–128.
[28] J. Ben Rosen, H. Park, J. Glick, Structured total least norm for nonlinear problems, SIAM J. Matrix Anal. Appl. 20 (1) (1998) 14–30.
[29] B.L. Satherley, C.R. Parker, Two-dimensional image reconstruction from zero sheets, Opt. Lett. 18 (1993) 2053–2055.
[30] G.W. Stewart, On the perturbation of LU, Cholesky, and QR factorizations, SIAM J. Matrix Anal. Appl. 14 (4) (1993) 1141–1145.
[31] J.R. Winkler, Structured matrix methods for the computation of multiple roots of a polynomial, J. Comput. Appl. Math. 272 (2014) 449–467.
[32] J.R. Winkler, The Sylvester resultant matrix and image deblurring, in: Lecture Notes in Computer Science (LNCS), vol. 9213, 2015, in press.
[33] J.R. Winkler, M. Hasan, A non-linear structure preserving matrix method for the low rank approximation of the Sylvester resultant matrix, J. Comput. Appl. Math. 234 (2010) 3226–3242.
[34] J.R. Winkler, M. Hasan, An improved non-linear method for the computation of a structured low rank approximation of the Sylvester resultant matrix, J. Comput. Appl. Math. 237 (1) (2013) 253–268.
[35] J.R. Winkler, M. Hasan, X.Y. Lao, Two methods for the calculation of the degree of an approximate greatest common divisior of two inexact polynomials, Calcolo 49 (2012) 241–267.
[36] J.R. Winkler, X.Y. Lao, M. Hasan, The computation of multiple roots of a polynomial, J. Comput. Appl. Math. 236 (2012) 3478–3497.
[37] C.J. Zarowski, X. Ma, F.W. Fairman, QR-factorization method for computing the greatest common divisor of polynomials with inexact coefficients, IEEE Trans. Signal Process. 48 (11) (2000) 3042–3051.
[38] Z. Zeng, The approximate GCD of inexact polynomials. Part 1: a univariate algorithm, preprint, 2004.