A New Constrained Total Variational Deblurring Model and Its Fast Algorithm

Bryan Michael Williams*, Ke Chen†, and Simon P. Harding‡

Abstract

Although image intensities are non-negative quantities, imposing positivity is not always considered in restoration models due to a lack of simple and robust methods of imposing the constraint. This paper proposes a suitable exponential type transform and applies it to the commonly-used total variation model to achieve implicitly constrained solution (positivity at its lower bound and a prescribed intensity value at the upper bound). Further to establish convergence, a convex model is proposed through a relaxation of the transformed functional. Numerical algorithms are presented to solve the resulting non-linear partial differential equations. Test results show that the proposed method is competitive when compared with existing methods in simple cases and more superior in other cases.

Key words. Total variation, image deblurring, alternating direction method of multipliers, box constraint, transforms.
AMS subject classifications. 68U10, 65J22, 65K10, 65T50, 90C25.

1 Introduction

Image processing techniques, such as image reconstruction which includes removing image noise from a given image (denoising) [29], reconstructing an image from a given blurred image (deblurring) [18], reconstructing the missing or damaged portion of an image (inpainting) [9], emphasizing the boundaries of an image by different filters or segmenting an image into subregions (segmentation) [17], have been widely used in many areas. Despite significant developments in photographic techniques and technology, blur is still a major cause for image quality degradation in clinical settings. This is due to many factors such as motion of the camera or more commonly in the case of retinal images the target scene, defocusing of the lens system, imperfections in the electronic, photographic, transmission medium, or obstructions. In this paper, we are concerned with variational models for restoration of such blurred and noisy images.

An observed blurred image can be written as a convolution of the true image with a blur function, known as the Point Spread Function (PSF) or kernel K [25]. There are three main deconvolution problems: 1) blind deconvolution, which includes the cases when both the kernel and the image are unknown [25, 26]; 2) semi-blind restoration, in which the kernel is assumed to belong to a class of parametric functions; or 3) non-blind deconvolution where only the image is unknown [18]. All three types are important not only in many scientific applications such as astronomical imaging, medical imaging, and remote sensing, but also for consumer photography.

*First two authors are with Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, The University of Liverpool, United Kingdom. Email: bryan@liv.ac.uk
†For correspondence, Email: k.chen@liv.ac.uk, Web: http://www.liv.ac.uk/cmit
‡St. Paul's Eye Unit, Royal Liverpool Hospitals, Liverpool, United Kingdom.

Deconvolution in the case of *known* blur, has been investigated widely in the last few decades giving rise to a variety of solutions [2, 3, 15, 22, 27, 24, 31, 30, 33]. In non-blind deconvolution, the point spread function is assumed known even though this information is not available in most of the real applications. In many cases, we know that our restored image must have strictly non-negative intensities, but the solution by traditional methods may yield results which are not necessarily positive. This has implications for most images with significant amounts of dark space, i.e. images with many pixel intensity values close to or equal to zero, as well as for blind deconvolution where the representation of certain blur functions has a significant amount of zero or near-zero values.

In this paper, we present a model for non-blind deconvolution which not only ensures a strictly positive result but also limits the upper boundary of the image intensity values, keeping them within a prescribed range. Related work in this area can be found as early as [7] and more work has been carried out in recent years which attempts to find strictly positive solutions for several applications, particularly astronomical imaging. Vogel and Bardsley [4] gave a method for large-scale minimization problems with non-negativity constraints using a cost functional including the statistics of the noise in the image data. A reduced Newton method was introduced such that Newton steps are only taken in the inactive variables, meaning those which are non-zero. A sparse matrix preconditioner was also introduced to improve convergence of Conjugate Gradient which is used to compute approximate reduced Newton steps. Benvenuto et al. [6] attempted to increase the efficiency of the projected Langweber method and iterative image space reconstruction algorithm, both of which demonstrate the property of semi-convergence. The results of the algorithms improve at the earlier iterations and then begin to worsen. The algorithms are also quite slow. The aim of Benvenuto et al. was primarily to improve the speed and convergence of these algorithms. The works of [12, 13] proposed other ideas based on nonnegative projections for deblurring. More recently, Chan et al. [15] gave a method for constrained image deblurring which is related to [4] but uses efficient alternate direction methods to drive the restored image closer to a projection of itself onto the ideal range. Since such projections (typically scaling or truncation) may cause a decrease in quality if simply applied at the end, the authors of [15] improve results by successively forcing the intensity values of the image to lie within a range which tends towards the ideal.

The rest of the paper is organised as follows. Section 2 reviews the total variation (TV) based variational models for denoising and deblurring. Section 3 presents our proposed transform and, consequently, its resulting model and algorithms. Section 4 discusses some refinement issues followed by Section 5 of numerical results and Section 6 of conclusions.

2 The TV based deblurring models

Noise and blur can be commonly found in digital images due to factors such as imperfections of the capturing equipment and scattering through nonhomogeneous medium. Following the work of [29], we consider the linear deblurring problem with additive noise

$$z = k * u + \eta \tag{1}$$

where z is the (known) observed image and η is the unknown noise function. There are two related models that one may consider.

Given knowledge of the blurring kernel k, the TV regularised model [29, 16, 4, 34] reconstructs u from solving

$$\min_u \int_\Omega (k * u - z)^2 d\Omega + \alpha_1 \|u\|_{TV}^\beta, \qquad \|u\|_{TV}^\beta = \int_\Omega |\nabla u|_\beta d\Omega, \tag{2}$$

where $\alpha_1 > 0$ and $\|u\|_{TV}^{\beta} = \int_\Omega |\nabla u|_\beta d\Omega = \int_\Omega \sqrt{u_x^2 + u_y^2 + \beta}\, d\Omega$, where β is a small non-negative constant, is a smooth approximation of the total variation. The model has been widely studied. Recently, algorithms for optimal selection of the parameter α_1 have also been proposed in [21, 37, 20].

Then to restore the kernel k from a known image u, a related model to (2) may be proposed

$$\min_u \int_\Omega (u * k - z)^2 d\Omega + \alpha_2 \|k\|_{TV}^{\beta}, \quad \text{s. t. } k \geq 0, \int_\Omega k(s,t) ds dt = 1, \tag{3}$$

where $\alpha_2 > 0$ and we have used the equality $u * k = k * u$. Our main concern in this paper is equation (2).

Here we remark that for (2), from our experience, the positivity method from [34] appears to be reliable. However, for model (3), the method of projecting solutions to satisfy the constraints $k \geq 0$, $\int_\Omega k(s,t) ds dt = 1$ seems less robust. Therefore, it is of importance to seek alternative and effective methods.

3 A transform based method for implicitly constrained reconstruction

In this section, we present a new transform method for imposing positivity for solving models (2)-(3). Our method will transform our constrained model to a non-constrained one. Therefore the positivity constraint is automatically satisfied. Below we use model (2) as the example.

Our motivation comes from a simple idea. If we wish for $u \geq 0$, we set $u = \exp(\psi)$ and reformulate our model in the new variable ψ. Then for any ψ, we can ensure $u \geq 0$. However, this seemingly great idea does not work because the inverse transform $\psi = \ln u$ does not allow $u = 0$. A remedial solution is to define the modified transform $u = \exp(\psi) - \epsilon$ so $\psi = \log(u + \epsilon)$; however to ensure $u \geq 0$, we require $\psi \geq \log(\epsilon)$ which implies that ψ must be constrained i.e. the underlying transform is not suitable. We would therefore aim to choose ϵ to be a very small positive number so that any final projection, if necessary, would have minimal effect on the result.

In order to impose a constraint on both the upper and lower bounds of u, we have found that a suitable exponential type transform is the following

$$u = \tilde{H}_\epsilon(\psi) = \frac{w + 2b}{1 + e^{-\frac{\psi}{\epsilon}}} - b$$

which resembles a smooth approximation to the Heaviside function given by

$$H(\psi) = \begin{cases} 0, & \text{if } \psi < 0 \\ 1, & \text{if } \psi \geq 0, \end{cases}$$

where $\epsilon, b, w > 0$, and $0 \approx -b \leq \tilde{H}_\epsilon(\psi) \leq w + b \approx w$ defines the intensity range for *any* ψ. Practically one may take, for (3), $b = 0.1, w = 255$ to accommodate the commonly used range $u \in [0, 255]$ and, for (27), $b = 0.01, w = 1$ to allow $k \in [0, 1]$. Note the inverse transform $\psi = -\frac{\epsilon}{2} \log \frac{w-u}{u+b}$ allows $u = 0$.

To allow generality, our proposed transform will be of the form

$$T(\psi) = \frac{a_1 + 2a_4}{1 + a_2 e^{\frac{-2\psi}{a_3}}} - a_4 \tag{4}$$

where $\mathbf{a} = (a_1, a_2, a_3, a_4)$ and all a_j's are positive. Note $0 \approx -a_4 \leq u = T(\psi) \leq a_1 + a_4 \approx a_1$ for any ψ. As illustrated in Fig.1, the generality allows us to adjust the maximal and minimal values of the range using a_1 and a_4, the spread of usable range of ψ using a_3 and the point of u at which ψ will be equal to zero using a_2. We can, if we wish, use this to restrict all values of ψ to positive but this is not necessary.

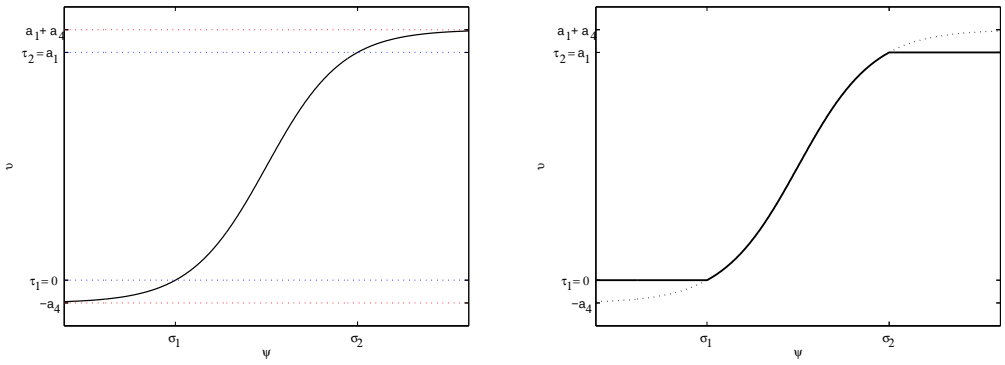


Figure 1: Graph of Heaviside Transform $u = T(\psi)$

Once the transform is specified, we now consider how to use it to reconstruct ψ first and hence the image u. The model (2) as studied in [34] can be transformed from

$$\min_u f(u) = \frac{1}{2}\Big\| \int k(x-x', y-y')u(x', y')d\Omega - z(x,y)\Big\|_2^2 + \alpha L(u)$$

(with $u \geq 0$) to the new problem for ψ

$$\min_\psi f(\psi) = \frac{1}{2}\Big\| \int k(x-x', y-y')T(\psi(x', y'))d\Omega - z(x,y)\Big\|_2^2 + \alpha L(T(\psi)) \tag{5}$$

where L denotes the TV regulariser for (2) and the H_1 for (9). The new and transformed model (5) has no constraint on ψ and yet can ensure (3) to have a positive solution u. However, since both terms in (5) are non-linear in ψ, it remains to address the numerical solution methods.

In what follows, we shall propose to treat term 1 in (5) by linearising $T(\psi)$ (due to the challenge associated with a non-local operator k) and term 2 by lagged diffusion ideas (as for solving the denoising [34]).

Linearisation of $T(\psi)$. The Taylor expansion of $T(\psi)$ about $\psi = 0$ is given by

$$T(\psi) = A + B\psi + O\left(\psi^2\right), \quad A = \frac{a_1 + 2a_4}{1 + a_2} - a_4, \; B = \frac{2a_2(a_1 + 2a_4)}{(1 + a_2)^2 a_3}.$$

Thus we can decompose $T(\psi)$ by separating its linear term in the form

$$u = T(\psi) = A + B\psi \; + \; \bar{\bar{v}}(\tilde{\psi}), \quad \bar{\bar{v}}(\tilde{\psi}) = \bar{v}(\tilde{\psi}) - A, \; \bar{v}(\tilde{\psi}) = T(\tilde{\psi}) - B\tilde{\psi}.$$

Iterative minimisation. Using the above decomposition, our solution strategy is as follows:

1: $u^{(0)} \leftarrow z$ and $\psi^{(0)} \leftarrow T^{-1}(u^{(0)})$
2: **for** Iterate on ℓ **do**

3: Solve for $\psi^{(\ell+1)}$, given $\psi^{(\ell)}$, from

$$\psi^{(\ell+1)} \leftarrow \min \|k * \psi^{(\ell+1)} B - \bar{z}(\psi^{(\ell)})\|_2^2 + \alpha \|B\psi^{(\ell+1)}) + \bar{v}(\psi^{(\ell)})\|_{TV}^{\beta} \qquad (*)$$

4: where $\bar{z}(\psi^{(\ell)}) = z - k * \bar{v}(\psi^{(\ell)})$.

5: **end for**

We now discuss how to solve the above equation (*) i.e.

$$\min_{\psi} \left\{ f(\psi) = \frac{1}{2} ||Bk * \psi - \bar{z}||_{L^2(\Omega)}^2 + \alpha \int_{\Omega} |\nabla \left(B\psi + \bar{v} \right)|_{\beta} d\Omega \right\}. \qquad (6)$$

Consider each term in turn. First let $f_1 = \frac{1}{2} ||Bk * \psi - \bar{z}||_{L^2(\Omega)}^2$ so $\min_{\psi} f_1$ is given when $\partial f_1 / \partial \psi = 0$. Here

$$\begin{aligned} \frac{\partial f_1}{\partial \psi} &= \frac{\partial}{\partial \psi} \frac{1}{2} ||Bk * \psi - \bar{z}||_{L^2(\Omega)}^2 \quad = \frac{1}{2} \frac{\partial}{\partial \psi} (Bk * \psi - \bar{z})^2 \\ &= \left(\frac{\partial}{\partial \psi} (Bk * \psi) \right) (Bk * \psi - \bar{z}) \quad = (Bk)^T (Bk * \psi - \bar{z}). \end{aligned}$$

Second let $f_2 = \int_{\Omega} \nabla (B\psi + \bar{v}) d\Omega$ and $\min_{\psi} f_2$ is given when $\dfrac{\partial}{\partial \epsilon} (f_2(\psi + \epsilon\phi))|_{\epsilon \to 0} = 0$ for an arbitrary function ϕ. We have

$$\begin{aligned} \frac{\partial}{\partial \epsilon} f_2(\psi + \epsilon\phi) \Big|_{\epsilon \to 0} &= \frac{\partial}{\partial \epsilon} \int_{\Omega} |\nabla (B(\psi + \epsilon\phi) + \bar{v})|_{\beta} d\Omega \Big|_{\epsilon \to 0} = \int_{\Omega} \frac{\partial}{\partial \epsilon} |\nabla (B(\psi + \epsilon\phi) + \bar{v})|_{\beta} d\Omega \Big|_{\epsilon \to 0} \\ &= \int_{\Omega} \frac{\nabla (B(\psi + \epsilon\phi) + \bar{v})}{|\nabla (B(\psi + \epsilon\phi) + \bar{v})|_{\beta}} \cdot \nabla B\phi d\Omega \Big|_{\epsilon \to 0} = \int_{\Omega} \frac{\nabla (B\psi + \bar{v})}{|\nabla (B\psi + \bar{v})|_{\beta}} \cdot \nabla B\phi d\Omega \\ &= -\int_{\Omega} \nabla \cdot \left(\frac{\nabla (B\psi + \bar{v})}{|\nabla (B\psi + \bar{v})|_{\beta}} \right) B\phi d\Omega + \int_{\Gamma} \frac{\nabla (B\psi + \bar{v})}{|\nabla (B\psi + \bar{v})|_{\beta}} \cdot B\phi \vec{n} d\Gamma. \end{aligned}$$

We have therefore that $\min_{\psi} \{ f = f_1 + f_2 \}$ is solved by

$$(Bk)^T (Bk * \psi - \bar{z}) + \alpha \nabla \cdot \left(\frac{\nabla (B\psi + \bar{v})}{|\nabla (B\psi + \bar{v})|_{\beta}} \right) B = 0 \qquad (7)$$

where $\bar{z} = \bar{z}(\psi) = z - k * \bar{v}(\psi)$ and $\bar{v} = \bar{v}(\psi) = T(\psi) - B\psi$.

Overall Algorithm. Assume u has a Dirichlet boundary condition. Then the discretised the Point Spread Function (PSF) k leads to a Block Toeplitz matrix with Toeplitz Blocks (BTTB) [22, 34]. In order to define the transform, we calculate the parameters $a_1, \ldots, a_4$ according to the Appendix. We calculate the initial estimate of $\psi^{(0)}$ given the initial estimate of $u^{(0)}$ as follows:

$$u = T(\psi) \quad = \quad \frac{a_1 + 2a_4}{1 + a_2 e^{-\frac{2\psi}{a_3}}} - a_4, \quad \psi = T^{-1}(u) = -\frac{a_3}{2} \ln \left(\frac{a_1 - u + a_4}{a_2 (u + a_4)} \right).$$

We then solve the Euler Lagrange equation (7) and finally transform the image back, obtaining our restored image u with positive entries. This is shown in Algorithm 1.

Algorithm 1 A Transform based algorithm for positivity

function TRANS($z, k, \alpha, \beta, \mathbf{a}, tol, maxit$)
  $u^{(0)} \leftarrow z$
  Calculate $\mathbf{a} = \{a_1, a_2, a_3, a_4\}$
  $\psi^{(0)} \leftarrow -(a_3/2)\log\left(\left(a_1 + a_4 - u^{(0)}\right) / \left(a_2(u^{(0)} + a_4)\right)\right)$
  for $\ell \leftarrow 1$ **to** $maxit$ **do**
    Solve equation for $\psi^{(\ell+1)}$ given $\psi^{(\ell)}$, i.e.

$$\psi^{(\ell+1)} \leftarrow \text{ SOLVE } (Bk)^T * \left(k * \psi^{(\ell+1)} - \overline{z}(\psi^{(\ell)})\right) - \alpha\nabla \cdot \frac{\nabla\left(B\psi^{(\ell+1)} - \overline{v}^{(\ell)}\right)}{\left|\nabla\left(B\widehat{\psi}^{(\ell+1)} - \overline{v}^{(\ell)}\right)\right|_\beta} = 0$$

    where $\overline{z}(\psi^{(\ell)}) = z - k * \overline{v}(\psi^{(\ell)})$ and $\widehat{\psi}$ denotes a lagging from ψ.
  end for
  On exit, $u^{(\ell+1)} \leftarrow (a_1 + 2a_4)/(1 + a_2\exp(-2\psi/a_3))$.
end function

4 Refinements and other solution strategies

4.1 Alternative Linearisation

In order to improve the speed of obtaining a solution, we carry out the Total Variation norm linearisation alongside the updating of the linearisation of the transform, thereby solving

$$(Bk)^T * \left(k * \psi^{(\ell+1)} - \bar{z}(\psi^{\ell})\right) - \alpha\nabla \cdot \frac{\nabla\left(B\psi^{(\ell+1)} - \bar{v}(\psi^{\ell})\right)}{\left|\nabla\left(B\psi^{(\ell)} - \bar{v}(\psi^{\ell})\right)\right|_\beta} = 0. \tag{8}$$

In this way, we hope to get speed-up due to the saving of iterations on $\widehat{\psi}$. Experimental results are shown in Figure 8 and error values and CPU times for this method and the previous transform method are given in Table 6. It can be noted that, the reduction in CPU time is significant.

4.2 Alternative Regularisation

While the total variation semi-norm which we have used in our model gives good results for images which have sharp changes in intensity and hence jumps in the pixel intensity value, improved results may be found by considering alternative regularisation to treat smooth images. In this section, we consider a simple form of alternative regularisation using the L_2 norm of the gradient of the image. More robust regularizations are based on high order regularisers; see [10, 8, 28, 19].

In the traditional case, using a least squares fitting term and L_2 as a regularisation term, we will obtain a linear partial differential equation to solve. We give this minimizing functional as

$$f(u) = \frac{1}{2}||k * u - z||^2_{L^2(\Omega)} + \frac{\alpha}{2}\int |\nabla u|^2 d\Omega. \tag{9}$$

The well-known Euler-Lagrange equation for the image u is therefore given by

$$k^T * (k * u - z) - \alpha\Delta u = 0. \tag{10}$$

Now referring to the above section, we substitute $u = B\psi + \bar{v}\left(\tilde{\psi}\right)$ to (9)

$$f(u) = \frac{1}{2}||k * (B\psi + \bar{v}\left(\tilde{\psi}\right)) - z||^2_{L^2(\Omega)} - \alpha \int |\nabla(B\psi + \bar{v}\left(\tilde{\psi}\right))|^2 d\Omega \tag{11}$$

$$= \frac{1}{2}||Bk * \psi - \bar{z}(\tilde{\psi})||^2_{L^2(\Omega)} - \alpha \int |\nabla(B\psi + \bar{v}\left(\tilde{\psi}\right))|^2 d\Omega \tag{12}$$

where $\bar{z}(\psi) = z - k * \bar{v}\left(\tilde{\psi}\right)$ and $\bar{v}\left(\tilde{\psi}\right) = T\left(\tilde{\psi}\right) - B\tilde{\psi}$. The linearised Euler-Lagrange equation is

$$k^T * \left(Bk * \psi - \bar{z}\left(\tilde{\psi}\right)\right) - \alpha\Delta\left(B\psi + \bar{v}\left(\tilde{\psi}\right)\right) = 0. \tag{13}$$

4.3 Initialisation of u and k

Since there exist many efficient algorithms for solving models (2) and (3) without the positivity constraints, one idea of acquiring good initialisations for u and k is through applying such algorithms first.

In fact, the simplistic L_2 method given by minimising (9) leads to solving the linear partial differential equation (10) which can be done efficiently. We may therefore use the solution of it as the initial estimate u and then our transform model will offer a positive solution.

As we shall see from the next section, for model (3) with the unknown kernel k, the Vogel's method [34] is no longer effective but we may use its result as an initial guess for our transform model; see Table 7 and Figure 9.

4.4 An Acceleration Algorithm for the Model

While our model performs well, it can often be rather slow to execute, particularly in cases of Gaussian blur. We address this issue using an alternating direction method (ADM) [15, 24, 36, 35]. We aim to separate our model into one of deblurring and one of denoising, each of which can be executed reasonable quickly. Starting with the unconstrained non-negative functional given by equation (5) we use the ADM to create the augmented Lagrangian functional

$$f(u, \psi, \lambda) = \frac{1}{2}||k * u - z||^2_{L^2(\Omega)} + \alpha L(T_{\mathbf{a}}(\psi)) + \frac{\gamma}{2}||u - T_{\mathbf{a}}(\psi)||^2_{L^2(\Omega)} + < \lambda, u - T_{\mathbf{a}}(\psi) > \tag{14}$$

where L represents either total variation (where we expect jumps in intensity) or L2 (where we expect smooth edges) i.e.

$$L(u) = \int_\Omega |\nabla u|_\beta d\Omega, \qquad \text{or} \qquad L(u) = \int_\Omega |\nabla u|^2 d\Omega. \tag{15}$$

Our aim is now to minimise f with respect to u, ψ and λ. Then we can give the Euler Lagrange equation for u:

$$k^T * (k * u - z) + \gamma\left(u - T_{\mathbf{a}}(\psi)\right) + \lambda = 0 \tag{16}$$

and, rearranging, we have

$$\left(k^T * k + \gamma\delta\right) * u = k^T * z + \gamma T_{\mathbf{a}}(\psi) - \lambda \tag{17}$$

where δ denotes the delta function and we can solve this using Fourier transforms. For additional support, we might add a term for u, given by $\chi L_1(u)$ where $\chi > 0$ and L_1 is a regularisation term. This model can be achieved by setting $\chi = 0$.

For the second equation, we minimise with respect to ψ as follows. We must deal with the nonlinearity of the transform. We do this by considering the Taylor expansion given by

$$T_{\mathbf{a}}(\psi) = A + B\psi + O(\psi^2)$$

and approximate the transform with $T_{\mathbf{a}}(\psi) = B\psi + R(\psi)$ where R, the residual, is given by $R = T_{\mathbf{a}}(\psi) - B\psi$. In practice, we will use this to form a fixed-point lagging technique by substituting $T_{\mathbf{a}}(\psi, \tilde{\psi}) = B\psi + R(\tilde{\psi})$, lagging $\tilde{\psi}$ and updating until $||\psi - \tilde{\psi}||$ is sufficiently small.

$$-B\lambda - \gamma B\left(u - (B\psi + \tilde{R})\right) + \alpha L(\tilde{\psi})\psi = 0 \tag{18}$$

where, for total variation,

$$L(\tilde{\psi})\psi = \frac{4\tilde{E}_1(a_1 + 2a_4)(\tilde{E}_1 - 1)|\nabla\psi|_\beta}{(1 + \tilde{E}_1)^3 a_3^2} - \nabla \cdot \left(\frac{2(a_1 + 2a_4)\tilde{E}_1}{(1 + \tilde{E}_1)^2 a_3 |\nabla\tilde{\psi}|_\beta}\nabla\psi\right).$$

Overall Algorithm

In order to solve our model, we begin with the initial estimate (typically the received image) and calculate the initial estimate of ψ using the chosen parameters. We then proceed to solve for u and ψ, updating λ. Our algorithm is given below in Algorithm 2.

Algorithm 2 An Accelerated Transform based algorithm for positivity

function ATRANS($z, k, \alpha, \beta, \gamma, \lambda^{(1)}, \mathbf{a}, tol, maxit$)
  $u^{(0)} \leftarrow z$
  Calculate $\mathbf{a} = \{a_1, a_2, a_3, a_4\}$
  $\psi^{(0)} \leftarrow -(a_3/2)\log\left(\left(a_1 + a_4 - u^{(0)}\right) / \left(a_2(u^{(0)} + a_4)\right)\right)$
  for $\ell_1 \leftarrow 1$ **to** $maxit$ **do**
    for $\ell_2 \leftarrow 1$ **to** $maxit$ **do**
      Solve equation for $u^{(\ell_2+1)}$ given $u^{(\ell_2)}$, i.e.

$$u^{(\ell_2+1)} \leftarrow \text{ SOLVE } k^T(ku^{(\ell_2+1)} - z) + \gamma(u^{(\ell_2+1)} - (B\psi + \tilde{R})) + \chi L_1(u^{\tilde{(\ell_2)}})u^{(\ell_2+1)} = -\lambda^{(\ell_1)}$$

    end for
    for $\ell_3 \leftarrow 1$ **to** $maxit$ **do**
      Solve equation for $\psi^{(\ell_3+1)}$ given $\psi^{(\ell_3)}$, i.e.

$$\psi^{(\ell_3+1)} \leftarrow \text{ SOLVE } \quad -\gamma B\left(u^{(\ell_2+1)} - (B\psi^{(\ell_3+1)} + \tilde{R})\right) + \alpha L(\psi^{\tilde{(\ell_3)}})\psi^{(\ell_3+1)} = B\lambda^{(\ell_1)}$$

    end for
    Update $\lambda^{(\ell_1+1)} \leftarrow \lambda^{(\ell_1)} + \gamma\left(u^{(\ell_2+1)} - T_{\mathbf{a}}\left(\psi^{(\ell_3+1)}\right)\right)$.
  end for
  On exit, $u^{(\ell_1+1)} \leftarrow (a_1 + 2a_4)/(1 + a_2\exp(-2\psi^{(\ell_3+1)}/a_3))$.
end function

4.5 A Reformulated Convex Model

We now wish to prove convergence of Algorithm 2. However, due to the lack of convexity of the model (14), this is not trivial. We therefore propose below a relaxation of this model so that the

new model is convex by the addition of a suitable term. We can then show convergence from the established approaches (see [23, 34, 5]). Tests in Section 5 will demonstrate that such a relaxation does not have a considerable impact on the solution or the quality of the restoration.

We aim to find an appropriate convex relaxation of this model by considering the fitting and regularisation terms separately since the sum of two convex functions is also convex. We attempt to obtain convexity of the fitting terms with the addition of a fitting term involving the function ψ of the form

$$\mu \int_\Omega (\psi - \zeta)^2 \, d\Omega$$

where ζ is a function not depending on ψ and μ is a non-negative real constant which must be sufficiently large to make the model (14) convex. In fact we see that, for this model, μ may be quite small so that assuming close proximity of the arguments this term should have only a small impact on the results. ζ should be a function which is approximately equal to ψ but not depend on u so that convexity with respect to u is unaffected. We take $\zeta = T_{\mathbf{a}}^{-1}(z^*)$ where

$$z^* = \arg\min_u \left\{ \int_\Omega (k * u - z)^2 \, d\Omega + \alpha \int_\Omega |\nabla u|^2 \, d\Omega \right\}.$$

Actually any other similar model that can be solved efficiently will also suffice.

The regularisation term requires a similar consideration for convexity, leading to

$$f(u, \psi; \lambda) = \frac{1}{2}||k * u - z||^2_{L^2(\Omega)} + \frac{\gamma}{2}||T_{\mathbf{a}}(\psi) - u||^2_{L^2(\Omega)} + < \lambda, T_{\mathbf{a}}(\psi) - u >$$

$$+\mu||\psi - \zeta||^2_{L^2(\Omega)} + \alpha \int_\Omega \left| \nabla \left(T_{\mathbf{a}}(\psi) + \theta||\psi - \zeta||^2_{L^2(\Omega)} \right) \right|_\beta d\Omega. \tag{19}$$

It turns out that μ and θ must satisfy

$$\mu \geq \frac{8\,(a_1 + 2a_4)}{27 a_3^2} \left(2\gamma(a_4 + \mathcal{L}_u) + \mathcal{L}_\lambda \right), \qquad \theta \geq -\frac{2(a_1 + 2a_4)(3\sqrt{3} - 5)}{\left(3 - \sqrt{3}\right)^3 a_3^2}. \tag{20}$$

To give an example of the values for the parameters, if we assume that our image is contained in the range $[0, 1]$, $\mathcal{L}_u = \mathcal{L}_\lambda = 0$, and $\mathbf{a} = \{1, 1, 0.44, 0.01\}$ for $u = T_{\mathbf{a}}(\psi)$, then $\mu \geq 0.04\gamma$ and $\theta \geq -1$.

In order to minimise the functional, we first calculate ζ and proceed with alternate minimisation. We present our overall algorithm below in Algorithm 3. For brevity, we do not present the Euler-Lagrange equation for ψ but it can be calculated in a similar manner to those above.

We would now like to show that the functional defined above is convex.

Theorem 4.1 *Let $\Omega \subset \mathbb{R}^n$ be a non-empty convex subset of $\mathbb{R}^n$ and $f : \Omega \to \mathbb{R} \cup \{+\infty\}$ be the function defined by (19–20). Then f is convex with respect to the argument ψ for ψ defined on Ω.*

Proof. It is sufficient to show that the functional (19) is a sum of two convex functions.
(i) The first part is given by

$$F(u, \psi) = \int_\Omega \gamma(T_{\mathbf{a}}(\psi) - u)^2 + \lambda(T_{\mathbf{a}}(\psi) - u) + \mu(\psi - \zeta)^2 \, d\Omega. \tag{21}$$

where μ must satisfy the above constraint and the second is given by the regularisation term.

Algorithm 3 A deblurring algorithm based on convex formulation

function CTRANS($k, z, \lambda^{(1)}; \alpha_s, \gamma, \alpha, \mu, \theta$)
  Solve the well known equation for ζ from

$$z^* = \min_u \left\{ f_s(u) = ||k * u - z||^2_{L^2(\Omega)} + \alpha_s ||\nabla u||^2_{L^2(\Omega)} \right\}$$

  Calculate $\mathbf{a} = \{a_1, a_2, a_3, a_4\}$
  $u^{(0)} \leftarrow z$
  $\psi^{(0)} \leftarrow T_{\mathbf{a}}^{-1}(z)$
  $\zeta \leftarrow T_{\mathbf{a}}(\mathbb{P}(z^*))$
  for $\ell \leftarrow 1$ **to** *maxit* **do**
    Solve equation for $u^{(\ell+1)}$ given $u^{(\ell)}$, i.e.

$$u^{(\ell+1)} \leftarrow \text{ SOLVE } \left((k^\dagger k + \gamma\delta) * u^{(\ell+1)} = k^\dagger z + \lambda^{(\ell)} + \gamma T_{\mathbf{a}} \left(\psi^{(\ell)} \right) \right)$$

    Solve equation for $\psi^{(\ell+1)}$ given $\psi^{(\ell)}$, i.e.

$$\psi^{\ell+1} \leftarrow \min_\psi \left\{ f\left(u^{(\ell+1)}, \psi; \lambda^{(\ell)} \right) \right\}$$

    Update $\lambda^{\ell+1} = \lambda^\ell + \gamma \left(u^{(\ell+1)} - T_{\mathbf{a}} \left(\psi^{(\ell+1)} \right) \right)$
  end for
  On exit, $u \leftarrow T_{\mathbf{a}} \left(\psi^{(\ell+1)} \right)$.
end function

To show that (21) is convex, we require the second order derivative given by

$$\frac{\partial^2 F(u,\psi)}{\partial \psi^2} = 2\mu - 2J(\psi) \left[-2\gamma (T_{\mathbf{a}}(\psi) + a_4) a_2 E - (2\gamma (T_{\mathbf{a}}(\psi) - u) + \lambda)(a_2 E - 1) \right] \tag{22}$$

to be non-negative, where $J(\psi) = \frac{2(T_{\mathbf{a}}(\psi)+a_4)a_2 E}{(1+a_2 E)^2 a_3^2}$ and $E = E(\psi) := \exp(-2\psi/a_3)$.

It is not difficult to show that the term to the right of $J(\psi)$ is contained in the bound $(-\infty, 2\gamma(a_4 + \mathcal{L}_u) + \mathcal{L}_\lambda)$ where $\mathcal{L}_u$ and $\mathcal{L}_\lambda$ are the lower bounds of u and λ respectively. For the function J, we can find that there is only one maximum by calculating the first derivative and finding the limits of the function as follows. We calculate the zero-point of the derivative

$$\frac{\partial J}{\partial \psi} = \frac{12(a_1 + 2a_4)a_2^2 E^2}{(1 + a_2 E)^4 a_3^3} - \frac{4(a_1 + 2a_4)a_2 E}{(1 + a_2 E)^3 a_3^3} = 0 \;\Leftrightarrow\; \psi = \frac{a_3}{2} \ln(2a_2)$$

at which the function J is non-negative and strictly positive assuming that at least one of a_1 and a_4 are non-zero, since $a_1, \ldots, a_4$ are non-negative constants.

Taking limits now and noting that $\lim_{\psi \to -\infty} E = \infty$ and $\lim_{\psi \to \infty} E = 0$, we find that the function J tends to 0 at $\pm\infty$ with a non-negative turning point given at $\psi = a_3 \ln(2a_2)/2$ which must be the maximum.

$$\lim_{\psi \to -\infty} \frac{2\left(\frac{a_1+2a_4}{1+a_2 E}\right) a_2 E}{(1 + a_2 E)^2 a_3^2} = \lim_{\psi \to -\infty} \frac{2(a_1 + 2a_4) a_2}{\left(\frac{1}{E} + 3a_2 + 3a_2^2 E + a_2^3 E^2\right) a_3^2} = 0 \tag{23}$$

$$\lim_{\psi \to \infty} \frac{2\left(\frac{a_1+2a_4}{1+a_2 E}\right) a_2 E}{(1 + a_2 E)^2 a_3^2} = 0. \tag{24}$$

Since the function tends to zero at both limits and has a single extremity, which is greater than or equal to zero, we can conclude that this is the maximum value and that the minimum is equal to zero, i.e.

$$J(\psi) \in \left(0, J\left(\frac{a_3}{2}\ln(2a_2)\right) = \frac{8\left(a_1+2a_4\right)}{27a_3^2}\right].$$

Substituting these bounds and inequalities, including μ from (20), into (22), it is clear that the convexity condition $\partial^2 F(u,\psi)/\partial\psi^2 \geq 0$ is satisfied.

(ii) For the (second part) total variation term, we begin by showing that if the function ω is convex then its total variation is also convex. It will then remain to show that the function (26) is convex given the restriction on the value θ. Recall the definition of a total variation via duality [14]

$$G(\psi) = G(\omega(\psi)) = \sup\left\{-\int_\Omega \omega(\psi)\mathrm{div}\phi\, dx : \phi \in C_c^\infty\left(\Omega;\mathbb{R}^N\right), |\phi(x)| \leq 1\, \forall x \in \Omega\right\}$$

and, when $\omega = \omega(\psi)$ is differentiable, $-\int_\Omega \omega(\psi)\mathrm{div}\phi\, dx = \int_\Omega \phi\cdot\nabla\omega(\psi)\, dx$. Letting $L_\phi : \psi \mapsto -\int_\Omega \omega(\psi)\mathrm{div}\phi\, dx$, we would like to show that if ω is convex then $G(\psi)$ is also convex. That is $\forall\ \psi_1, \psi_2$ and $t \in [0,1]$, we have $G(t\psi_1 + (1-t)\psi_2) \ \leq\ tG(\psi_1) + (1-t)G(\psi_2)$. Assuming that $\omega(\psi)$ is convex with respect to ψ then we have the relation

$$\omega(t\psi_1 + (1-t)\psi_2) \ \leq\ t\omega(\psi_1) + (1-t)\omega(\psi_2)$$

and

$$L_\phi(t\psi_1 + (1-t)\psi_2) \ \leq\ tL_\phi(\psi_1) + (1-t)L_\phi(\psi_2) \ \leq\ tG(\psi_1) + (1-t)G(\psi_2) \tag{25}$$

Since G is the supremum of the functions L_ϕ, i.e.

$$\sup_\phi L_\phi(t\psi_1 + (1-t)\psi_2) = G(t\psi_1 + (1-t)\psi_2),$$

we have by (25) that $G(t\psi_1 + (1-t)\psi_2) \leq tG(\psi_1) + (1-t)G(\psi_2)$. That is, if the transform $\omega(\psi)$ is convex for ψ then the total variation is convex for ψ. It remains to show that the function

$$\omega(\psi) = T_{\mathbf{a}}(\psi) + \theta||\psi - \zeta||^2_{L^2(\Omega)}, \tag{26}$$

where ζ is as described above, is convex. Proceeding as in (i), we calculate the second derivative

$$\frac{\partial^2\omega}{\partial\psi^2} = 2\theta - 2J_1(\psi), \qquad J_1(\psi) := \frac{2(a_1+2a_4)a_2E(1-a_2E)}{(1+a_2E)^3a_3^2}.$$

We would like to find the upper bound of this function. We consider the limits

$$\lim_{\psi\to-\infty} J_1(\psi) = \lim_{\psi\to-\infty}\left(\frac{2(a_1+2a_4)a_2E(1-a_2E)}{(1+a_2E)^3a_3^2}\right) = 0,$$

$$\lim_{\psi\to\infty} J_1(\psi) = \lim_{\psi\to\infty}\left(\frac{2(a_1+2a_4)a_2E(1-a_2E)}{(1+a_2E)^3a_3^2}\right) = 0,$$

which are equal to zero. We now find the extrema

$$\frac{\partial J_1}{\partial\psi} \ =\ -8(a_1+2a_4)a_2E\frac{a_2^2E^2 - 4a_2E + 1}{a_3^3(1+a_2E)^4} \ =\ 0 \ \Leftrightarrow\ \psi \ =\ \frac{-a_3}{2}\frac{2\pm\sqrt{3}}{a_2}$$

at which J_1 is given by

$$J_1\left(\frac{-a_3}{2}\frac{2\pm\sqrt{3}}{a_2}\right) = -\frac{2(a_1+2a_4)(2\pm\sqrt{3})\left(1\pm\sqrt{3}\right)}{\left(3\pm\sqrt{3}\right)^3 a_3^2}.$$

It is easy to observe that a positive value is obtained at $\psi = -a_3(2-\sqrt{3})/2a_2$ and a negative value is obtained at $\psi = -a_3(2+\sqrt{3})/2a_2$. We can therefore conclude that the values of J_1 lie in the range

$$\left[-\frac{2(a_1+2a_4)(3\sqrt{3}+5)}{\left(3+\sqrt{3}\right)^3 a_3^2}, \frac{2(a_1+2a_4)(3\sqrt{3}-5)}{\left(3-\sqrt{3}\right)^3 a_3^2}\right],$$

so that $\partial^2\omega(\psi)/\partial\psi^2 = 2\theta - 2J_1(\psi) \geq 0$, if θ is from (20), as required. ∎

5 Experimental results

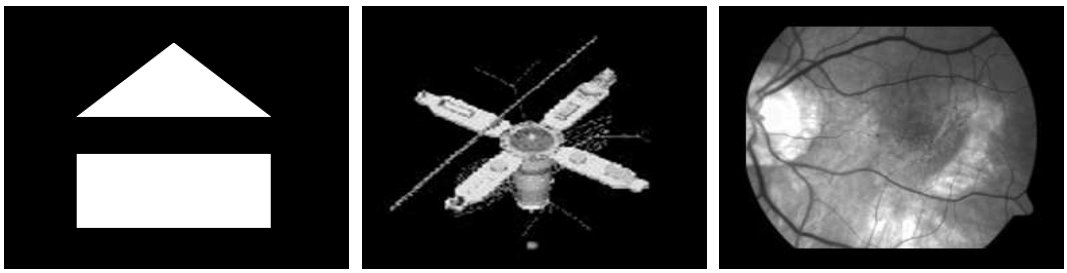
Our experimental tests are hoped to show the effectiveness of image restoration by our Algorithm 1 in comparison with Vogel's positivity method [4, 34], the projection method [15] and other methods that do not impose positivity constraints. We also compare with unconstrained (and partly constrained) models which have the constraint applied at the end by truncation or scaling. Specifically, in tables and figures, we denote the compared methods by these abbreviations:

- ROF: the well-known model (2) without positivity constraint.
- ROF$_{Thr}$: the well-known model (2) with positivity and upper limit constraints applied at the end by truncation.
- ROF$_{Sca}$: the well-known model (2) with positivity and upper limit constraints applied at the end by scaling.
- Vogel: the non-negatively constrained restoration model by [4].
- Vogel$_{Thr}$: the non-negatively constrained restoration model by [4] with upper limit constraint applied at the end by truncation.
- Vogel$_{Sca}$: the non-negatively constrained restoration model by [4] with upper limit constraint applied at the end by scaling.
- Proj: the constrained projection model by [15].
- New1: Algorithm 1 for model (5).
- New1L$_2$: Algorithm 1 to solve the minimization of (9).
- New2: Algorithm 2 for model (14) i.e. an accelerated version of New1.
- MixL$_2$TV: Algorithm 1 to solve (13) followed by Algorithm 1 to solve (7) using the solution of (13) as the initial estimate.
- MixVogTV: Algorithm 1 to solve (7) using the solution given by Vogel as the initial estimate.

- New3: Algorithm 3 for the reformulated model (19) i.e. the convex version of New2.

We use "Received" to mean the received image z.

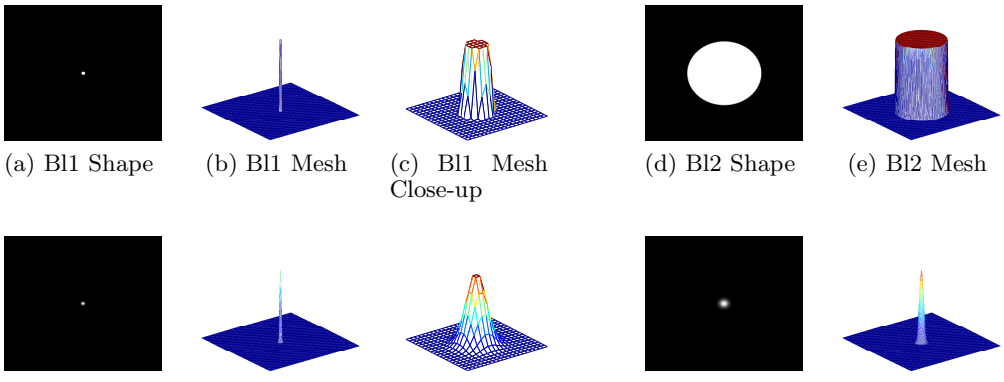
Seven sets of experimental results using 3 test images: the box-triangle image (Im1), the satellite image (Im2) and the retina image (Im3) are selected; see Figure 2. For the transform $u = T(\psi)$, we choose $a_1 = 1, 1.08, 255$ and $a_4 = 10^{-2}, 10^{-2}, 0.5$ respectively for the 3 test images (note a_2, a_3 are set as in Appendix). For the blurring model (1), we have considered small and large levels of motion blur



(a) Im1 - Box-Triangle Image (b) Im2 - Satellite Image (c) Im3 - Retina Image

Figure 2: Test case images.

(Bl1 and Bl2 respectively) and small and large levels of Gaussian blur (Bl3 and Bl4 respectively); see Figure 3.



(a) Bl1 Shape (b) Bl1 Mesh (c) Bl1 Mesh Close-up (d) Bl2 Shape (e) Bl2 Mesh

(f) Bl3 Shape (g) Bl3 Mesh (h) Bl3 Mesh Close-up (i) Bl4 Shape (j) Bl4 Mesh

Figure 3: PSFs used for test cases. Images (a)-(c) show Bl1 - small motion blur, images (d)-(e) show Bl2 - large motion blur, images (f)-(h) show Bl3 - small Gaussian blur, and images (i)-(j) show Bl4 - large Gaussian blur.

There are several common measures for testing the quality of the restored image, including the following. We let u_{true} denote the true image, u the restored image, z the received image and let m and n be the number of pixels horizontally and vertically respectively. Then we have:

- Mean Squared Error (MSE) is given by $MSE = \frac{1}{mn}\sum_{x,y}(u_{\text{true}}(x,y) - u(x,y))^2$ and Root Mean Squared Error (RMSE) is given by $RMSE = \sqrt{MSE}$.
- Signal-to-Noise Ratio (SNR) in dB is given by $SNR = 10\log_{10}\left(\frac{\sum_{x,y}|u_{\text{true}}(x,y)|^2}{\sum_{x,y}|u_{\text{true}}(x,y)-u(x,y)|^2}\right)$
- Peak Signal-to-Noise Ratio (PSNR) is given by $PSNR = 20\log_{10}\left(\frac{\max_{x,y}|u_{\text{true}}(x,y)|}{RMSE}\right)$.

Note that the RMSE is given by the L_2 norm of the difference between the true image and the restored image divided by the total number of pixels, i.e. $RMSE = (1/mn)||u_{\text{true}} - u||_{L^2(\Omega)}$. Given astronomical images and images with significant amounts of black space, it is typically more common to use the L1 norm. We expect that these may provide more accurately descriptive measures of our data and the impact of the model in terms of non-negativity. We therefore propose the measures

- L1 Error given by

$$Er_1 = ||u_{\text{true}} - u||_{L^1(\Omega)} = \frac{1}{mn} \sum_{x,y} |u_{\text{true}}(x,y) - u(x,y)|.$$

- A version of PSNR using the L_1 norm of the difference between the true image and the restored image is given by

$$Er_2 = 20 \log_{10} \left(\frac{\max_{x,y} |u_{\text{true}}(x,y)|}{Er_1} \right).$$

Model (1) with Gaussian blur. Result set 1 uses Im1 corrupted by Gaussian blur to demonstrate the effectiveness of the model in keeping the intensity values of the image constrained. We see in Figure 4 and Table 1 that [4] keeps the image positive but allows some points to take intensity values which are outside of the expected range, while [15] and the new models successfully keep the intensity values positive and within the expected range at all points.

Model (1) with Motion blur. Result set 2 consists of Im2 and Im3 corrupted by small motion or small Gaussian blur. We see in Figure 5 and Tables 2–3 that for images corrupted by small levels of blur the results are competitive between the models. Error values are improved but visual quality is similar.

Model (1) with Heavy blurs. Result set 3 consists of Im2 corrupted by larger levels of blur (Bl2 and Bl4). We see in Figure 6 and Table 4 that that results are improved visually and in the error values for the new model in the case of Bl2. For Bl4, the Transform Model appears to be a closer approximation but the error values are similar.

Model (1) with Blur and a varying level of noise. Result set 4 consists of Im2 corrupted by Bl3 and varying amounts of noise (1% and 50%). We see in Figure 7 and Table 5 that visually the Transform model offers some improvement in quality while the error values are similar.

Model (1) by Algorithm 1 with alternative linerisation (8). Result set 5 shows in Figure 8 and Table 6 the results using the linearised Transform model. We can see that for the same quality of the restored image, the CPU time is improved.

Algorithm 1 combined with Vogel's model. Result set 6 shows in Figure 9 and Table 7 examples using the received image as the initial estimate and the results of Vogel's model as the initial estimate. We can see that this technique is useful for restoring the PSF given the image. In the case of the motion blur example, the CPU time is significantly improved and in the case of Gaussian blur, the error value is improved. In all cases, the visual quality is adequate.

Model (2) with Blurs. Now we consider the solution of model (3) for k. Result set 7 consists of motion and Gaussian blur PSFs which are regarded as being blurred by Im2. The task here is to recover the PSF given the true image. As the initial estimate, rather than taking the received data z as the initial estimate (since it is not expected to be a good approximation of the true kernel) we make an estimate of the kernel based on observation of the received data. We see in Figure 10 that in both cases, each of the models are able to obtain good approximations of the kernel, however ROF is unable to retain non-negativity in both cases and Vogel, while successfully ensuring positivity of the approximated kernel, struggles to get correct smaller values as well as larger values whereas the transform model is able keep the values close to zero as well as ensuring positivity of the result.

Finally, to simultaneously restore both u and k in the so-called blind deconvolution problem, the TV based model by [18] is the following

$$\min_u \int_\Omega (u * k - z)^2 d\Omega + \alpha_1 \|u\|_{TV}^\beta + \alpha_2 \|k\|_{TV}^\beta, \quad \text{s. t. } k \geq 0, \int_\Omega k(s,t) ds dt = 1, \tag{27}$$

where $\alpha_1, \alpha_2 > 0$. Related studies can be found in [1, 11, 36, 32, 38]. In other experiments, we have tried double transforms which appear to improve the robustness. This model will be investigated further in the future.

Method	cpu	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0771	39.35	16.27	15.79	22.26	n/a
ROF	32.2	0.0589	27.39	19.41	18.33	24.60	2.33
ROF_{Thr}	32.2	0.0528	26.40	19.73	19.15	25.54	3.28
ROF_{Sca}	32.2	0.1573	55.28	13.31	8.35	16.07	-6.20
Vogel	37.8	0.0320	17.24	23.43	23.61	29.89	7.63
Vogel_{Thr}	37.8	0.0303	17.24	23.43	24.02	30.36	8.10
Vogel_{Sca}	37.8	0.0765	25.04	20.19	14.75	22.33	0.07
Proj	33.5	0.0378	18.78	22.69	22.12	28.46	6.20
New1	59.7	**0.0149**	6.86	31.43	**30.28**	**36.55**	**14.29**
New2	26.3	0.0236	11.31	27.10	26.25	32.53	10.26
New3	15.8	0.0241	**4.45**	**35.20**	26.02	32.35	10.09

Table 1: Result Set 1 - Error values for Im1 corrupted by Gaussian blur with no Noise. We can see that the error values are improved when using the Transform models and CPU time is improved by using New2–New3. As designed, the results of New2–New3 are very similar, showing that the additional term does not have a considerable effect on results.

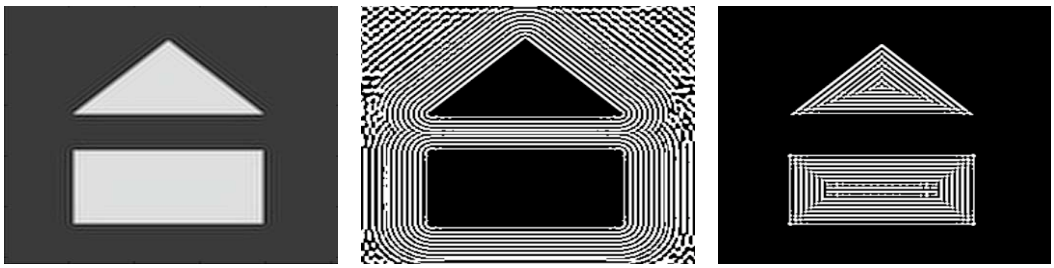
6 Conclusion and Future Work

We have presented models to reconstruct images and PSFs and demonstrated that they can ensure positivity through introducing a transform and also keep the intensities of the restored data within the appropriate range. We have also demonstrated that the model offers competitive results in the case of small levels of blur and noise but much improved results in the case of corruption by larger levels of blur and noise. This model is particularly effective in giving a close approximation of the kernel (in the case where the image is known) which is of great importance in the case of blind deblurring. The transform idea is applicable potential to a class of other variational models. Since non-negativity is a significant criterion for blind deblurring models, we hope to consider such applications in the near future.

Appendix – Selection of Parameters in $T(\psi)$

The parameter a_1 is easily chosen, assuming knowledge of the bits-per-sample (bps) value of the true image and the blurred image. This will typically be between 1 and 255 for images of bps 1 to 8 respectively, but can be quite low for the kernel. For example, a fairly compact-radius out-of-focus blur may have a kernel value upper limit of 10^{-2}. While a larger value of a_1 should still give a good approximation, it is essential that a_1 be at least as large as the maximum image intensity value or kernel value and advisable that it be close to this. The parameter a_4 should be chosen in proportion to a_1. Typically, $a_4 = a_1/255$ is a sensible value.

(a) True Image (b) Blur Kernel (c) Blurred Image

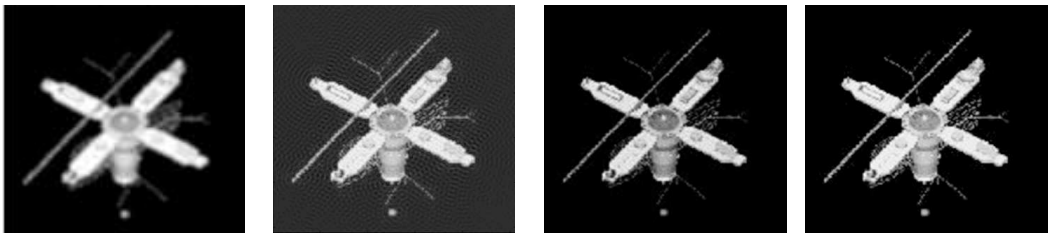


(d) ROF Restored (e) ROF Negative (f) ROF $> \zeta$

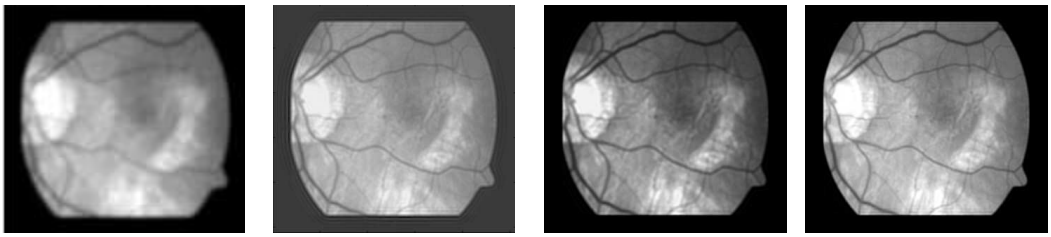
(g) Vogel Restored (h) Vogel Negative (i) Vogel $> \zeta$

(j) New1 Restored (k) New1 Negative (l) New1 $> \zeta$

Figure 4: Result Set 1: Restoring Im1 corrupted by Bl3 with no noise. From top to bottom, we have: 1) the true image, kernel, and corrupted data; 2) the result using the ROF method; 3) the result using Vogel's method; 4) the result using the Transform method. From left to right, we have (on rows 2-4): 1) the restored image; 2) the negative values of the restored image in white; 3) the points where the intensity values are greater than the expected upper limit in white. Note that the Transform method and Vogel's method can both ensure positivity but the transform method can control the upper bound of the intensity range.

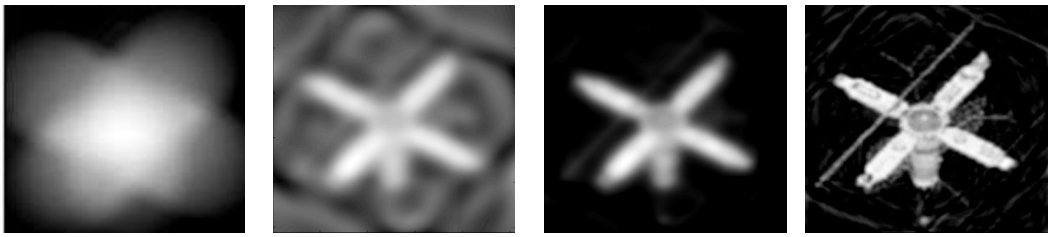


(a) Blurred Image (b) ROF Restored (c) Vogel Restored (d) New1 Restored



(e) Blurred Image (f) ROF Restored (g) Vogel Restored (h) New1 Restored

Figure 5: Result Set 2 - restoring images Im2 and Im3 corrupted by small motion blur Bl1 or small Gaussian blur Bl3. In some cases the results from the Transform model appear sharper than other models and more small detail is visible.



(a) Blurred Image (b) ROF Restored (c) Vogel Restored (d) New1 Restored

(e) Blurred Image (f) ROF Restored (g) Vogel Restored (h) New1L_2 Restored

Figure 6: Result Set 3 - Restoring Im2 corrupted by Bl2 (top line) and by Bl4 (bottom line). We can see a significant improvement in the result from the Transform method in the case of corruption by Bl2, and the results are competitive in the case of Bl4.

Error values for Im2 corrupted by Bl1							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0478	11.18	27.19	14.47	26.41	n/a
ROF	40.6	0.0211	6.31	32.17	21.93	33.51	7.10
ROF_{Thr}	40.6	0.0183	5.05	34.09	23.15	34.74	8.33
ROF_{Sca}	40.6	0.0932	25.81	19.93	9.76	20.62	-5.79
Vogel	31.9	0.0107	3.08	38.39	27.79	39.39	12.98
$\text{Vogel}_{\text{Thr}}$	31.9	0.0107	3.08	38.39	27.79	39.39	12.98
$\text{Vogel}_{\text{Sca}}$	31.9	0.0478	3.12	38.27	27.85	39.38	12.98
Proj	16.2	0.0054	1.37	45.40	33.76	45.33	18.92
New1	38.7	0.0036	**0.96**	**48.54**	37.31	48.88	22.47
New2	12.8	0.0051	1.36	45.51	34.24	45.81	19.40
New3	12.6	**0.0033**	1.16	46.89	**38.27**	**49.75**	**23.34**
Error values for Im3 corrupted by Bl1							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0362	19.19	22.50	21.39	28.82	n/a
ROF	34.7	0.0178	9.09	29.00	27.62	34.97	6.16
ROF_{Thr}	34.7	0.0164	8.52	29.55	28.33	35.69	6.88
ROF_{Sca}	34.7	0.0557	27.47	19.39	17.70	25.08	-3.74
Vogel	25.2	0.0084	3.95	36.24	34.14	41.51	12.70
$\text{Vogel}_{\text{Thr}}$	25.2	0.0084	3.95	36.24	34.14	41.52	12.70
$\text{Vogel}_{\text{Sca}}$	25.2	0.0113	4.54	35.02	31.39	38.90	10.09
Proj	17.5	0.0056	1.86	42.79	37.62	44.97	16.16
New1	66.0	**0.0020**	0.94	48.67	**46.57**	**54.03**	**25.21**
New2	12.9	0.0044	1.30	45.86	39.69	47.04	18.23
New3	15.0	0.0027	**0.80**	**50.07**	44.03	51.34	22.52

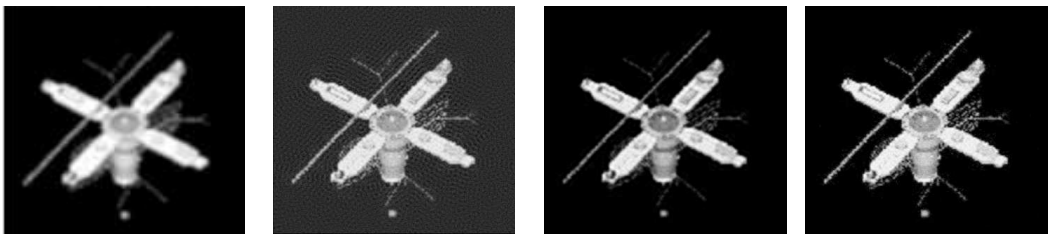
Table 2: Result Set 2 - Error values for images Im2 and Im3 corrupted by Bl1. It can be noticed that error values are improved using the Transform models. While CPU time is higher than that of competing models, New2–New3 can reduce CPU time while retaining similar or improved PSNR. As designed, the results of New2–New3 are very similar, showing that the additional term does not have a considerable effect on results.

Error values for Im2 corrupted by Bl3							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0562	12.72	26.08	12.96	25.01	n/a
ROF	36.4	0.0263	7.01	31.25	19.99	31.59	6.59
ROF_{Thr}	36.4	0.0249	6.51	31.89	20.46	32.07	7.06
ROF_{Sca}	36.4	0.1221	33.49	17.67	7.52	18.27	-6.74
Vogel	32.2	0.0233	6.38	32.07	21.01	32.65	7.64
$\text{Vogel}_{\text{Thr}}$	32.2	0.0233	6.38	32.07	21.01	32.65	7.64
$\text{Vogel}_{\text{Sca}}$	32.2	0.0236	6.31	32.16	21.07	32.55	7.54
Proj	17.4	0.0203	5.52	33.33	22.27	33.87	8.86
New1	45.8	**0.0142**	**4.18**	**35.73**	**25.39**	**36.97**	**11.96**
New2	13.0	0.0172	5.05	34.10	23.70	35.28	10.27
New3	16.2	0.0156	4.70	34.73	24.58	36.13	11.12
Error values for Im3 corrupted by Bl3							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0422	24.01	20.56	20.04	27.49	n/a
ROF	35.2	0.0236	14.79	24.76	25.17	32.54	5.05
ROF_{Thr}	35.2	0.0226	14.35	25.03	25.56	32.93	5.44
ROF_{Sca}	35.2	0.0855	42.65	15.57	14.18	21.36	-6.12
Vogel	23.8	0.0169	9.77	28.37	28.06	35.44	7.95
$\text{Vogel}_{\text{Thr}}$	23.8	0.0169	9.77	28.37	28.08	35.45	7.96
$\text{Vogel}_{\text{Sca}}$	23.8	0.0240	10.98	27.36	24.68	32.39	4.90
Proj	25.1	0.0177	11.44	26.99	27.68	35.05	7.56
New1	87.4	**0.0127**	**7.84**	**30.27**	**30.53**	**37.89**	**10.40**
New2	13.1	0.0171	11.04	27.31	27.99	35.35	7.87
New3	16.3	0.0147	9.10	28.99	29.35	36.66	9.17

Table 3: Result Set 2 - Error values for images Im2 and Im3 corrupted by Bl3. It can be noticed that error values are improved using the Transform model. While CPU time is higher than that of competing models, New2–New3 can reduce CPU time without a significant reduction in PSNR. As designed, the results of New2–New3 are very similar, showing that the additional term does not have a considerable effect on results.

Error values for Im2 corrupted by Bl2							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.22	63.17	12.80	-2.13	13.66	n/a
ROF	2.01	0.13	33.77	18.24	6.08	18.31	4.65
Vogel	16.02	0.11	26.55	20.33	7.30	19.71	6.05
New1	47.64	**0.06**	**14.87**	**25.36**	**14.03**	**25.41**	**11.75**
New1L_2	30.18	0.07	18.68	23.38	11.77	23.33	9.67
MixL_2TV	13.65	0.10	24.32	21.09	9.03	20.87	7.21
Error values for Im2 corrupted by Bl4							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0909	21.40	21.56	8.04	20.83	n/a
ROF	54.8	0.0596	14.98	24.66	12.72	24.49	3.66
Vogel	37.9	0.0565	13.00	25.88	13.11	24.96	4.13
New1	31.3	**0.0489**	**11.72**	**26.78**	**14.45**	**26.22**	**5.39**

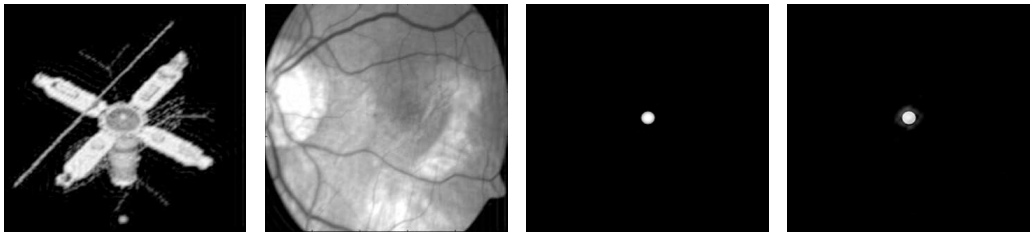
Table 4: Result Set 3 - Error values for Im2 corrupted by Bl2 and Bl4. There is a noticeable improvement in the case of and while the results for Bl4 are competitive, the transform is slightly improved over competing models.



(a) Blurred Image (b) ROF Restored (c) Vogel Restored (d) New1 Restored

(e) Blurred Image (f) ROF Restored (g) Vogel Restored (h) New1 Restored

Figure 7: Result Set 4 - Restoring Im2 corrupted by Bl3 and 1% noise (top row) and 50% noise (bottom row). We can see that visually the Transform method appears to give improved results for weaker and stronger levels of noise.



(a) Im2 (b) Im3 (c) Bl1 (d) Bl3

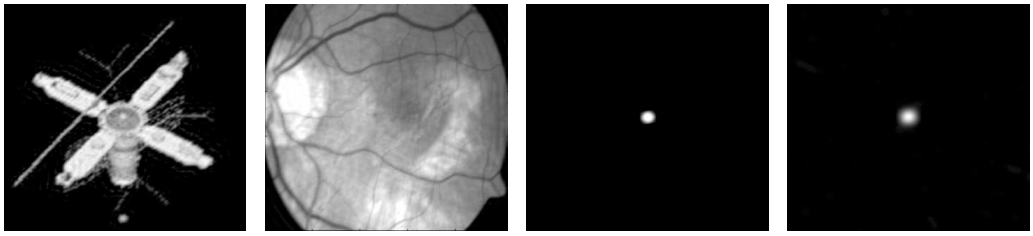
Figure 8: Result Set 5: Restored images and PSFs using the Linearised Transform method. The received data from which Im2 and Im3 were restored was corrupted by Bl1, and the received data from which Bl1 and Bl3 were restored was corrupted by Im2. We can see that the linearisation does not affect the visual quality significantly.

Error values for Im2 corrupted by Bl3 and 1% noise.							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0479	11.34	27.07	14.49	26.40	n/a
ROF	42.1	0.0304	7.85	30.26	18.82	30.35	3.95
Vogel	12.1	0.0237	6.41	32.03	20.80	32.52	6.12
New2	4.9	**0.0196**	**5.73**	**33.01**	**22.61**	**34.15**	**4.85**
Error values for Im2 corrupted by Bl3 and 50% noise.							
Method	CPU Time	rmse	Er_1	Er_2	snr	psnr	isnr
Received	n/a	0.0639	60.71	13.14	-13.24	12.59	n/a
ROF	15.00	0.0783	19.58	22.97	11.50	22.77	10.19
Vogel	5.64	0.0980	22.91	21.61	9.72	20.86	8.27
New1	55.76	**0.0718**	**17.12**	**24.14**	**12.20**	**23.52**	**10.94**

Table 5: Result Set 4 - Error values for Im2 corrupted by Bl2 and varying amounts of noise. We can see that the Transform model can offer improved results, particularly for larger levels of noise.

	Transform Model (New1)		Linearised New1	
Image	psnr	CPU Time	psnr	CPU Time
Im2	30.32	60.07	30.54	**34.10**
Im3	35.62	83.05	35.51	**35.09**
Bl1	38.63	72.25	38.24	**51.00**
Bl3	39.56	82.77	37.59	**47.84**

Table 6: Result Set 5: Error values and CPU time for restoring images Im2 and Im3 as well as PSFs BL1 and BL3 using the Transform method and the Linearised Transform method. We can see that the quality of the restored image is not significantly different for each case but the CPU time is improved using the Linearised Transform method.

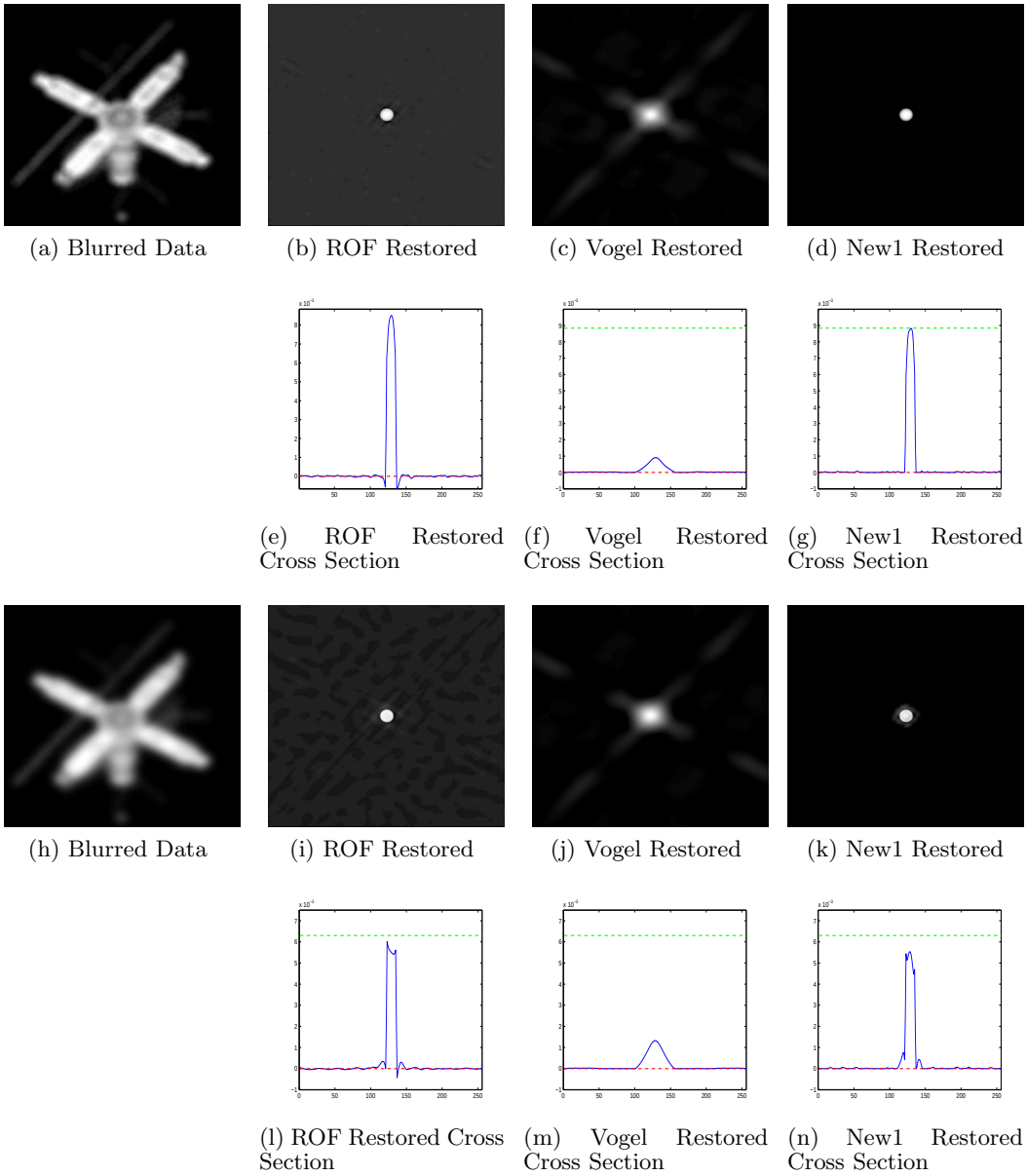


(a) Im2 (b) Im3 (c) Bl1 (d) Bl3

Figure 9: Result Set 6: Restored images and PSFs using the Linearised Transform method with the result of Vogel's method as the initial estimate.

	New1		MixVogTV	
Image	psnr	CPU Time	psnr	CPU Time
Im2	30.54	**34.10**	**30.61**	39.79
Im3	35.51	**35.09**	**35.71**	46.83
Bl1	38.24	51.00	**38.80**	**27.59**
Bl3	37.59	**47.84**	**42.53**	51.14

Table 7: Result Set 6: Error values and CPU times for restoring images Im2 and Im3 and PSFs Bl1 and Bl3 using the Linearised Transform method with the received data z as the initial estimate (New1) and the result from Vogel's method as the initial estimate (MixVogTV). The CPU time is rarely lower when using the closer initial estimate but the image quality is improved in all cases.



(a) Blurred Data (b) ROF Restored (c) Vogel Restored (d) New1 Restored

(e) ROF Restored Cross Section (f) Vogel Restored Cross Section (g) New1 Restored Cross Section

(h) Blurred Data (i) ROF Restored (j) Vogel Restored (k) New1 Restored

(l) ROF Restored Cross Section (m) Vogel Restored Cross Section (n) New1 Restored Cross Section

Figure 10: Result Set 7 - Restoring Bl1 (1st and 2nd rows) and Bl2 (3rd and 4th rows) corrupted by Im1 restored using TV restoration (ROF), Vogel's model (Vogel) and the transform model (New1). In the cross-section images, the blue line is the restored image, the red dashed line is the lower bound of the true blur function and the green dashed line is the upper bound of the true blur function. Of the three approximations, as demonstrated in the cross-section images on the 2nd and 4th rows, the TV model gives many negative values in the approximation both kernels, and Vogel's model has no negative values but struggles to get a close approximation while the transform model does a good job.

We attempt to select the remaining parameters a_2 and a_3 in order to control the upper and lower bounds of ψ as well as the value of ψ when u is equal to zero. In order to control the bounds, we define a length $\Sigma = \sigma_4 - \sigma_3$ where σ_3 and σ_4 represent two intensity values of ψ. We would then like for $\tau_4 - \tau_3 = T(\sigma_4) - T(\sigma_3) = \Sigma$. From $\psi(\tau) = T^{-1}(\tau) = -\frac{a_3}{2}\ln\left(\frac{a_1-\tau+a_4}{a_2(\tau+a_4)}\right)$, we have

$$\Sigma = \sigma_4 - \sigma_3 = \psi(\tau_4) - \psi(\tau_3) \tag{28}$$

$$= \frac{a_3}{2}\ln\left(\frac{(a_1-\tau_3+a_4)(\tau_4+a_4)}{(\tau_3+a_4)(a_1-\tau_4+a_4)}\right). \tag{29}$$

So, assuming we fix Σ, τ_3, τ_4, a_1 and a_4, we have

$$a_3 = \frac{2\Sigma}{\ln\left(\frac{(a_1-\tau_3+a_4)(\tau_4+a_4)}{(\tau_3+a_4)(a_1-\tau_4+a_4)}\right)}$$

For our model, we fix the width $\Sigma = \tau_4 - \tau_3$ (see Figure 11) and let $\tau_4 = a_1 - \tau_3$. Then, from 29, we have

$$a_3 = \frac{2(\tau_4-\tau_3)}{\ln\left(\frac{(a_1-\tau_3+a_4)(\tau_4+a_4)}{(\tau_3+a_4)(a_1-\tau_4+a_4)}\right)} = \frac{a_1-2\tau_3}{\ln\left(\frac{(a_1-\tau_3+a_4)}{(\tau_3+a_4)}\right)}.$$

The only remaining parameter which a_3 is dependent on and which has not already been decided is τ_3. We find that $\tau_3 = a_1/4$ is adequate for the transform.

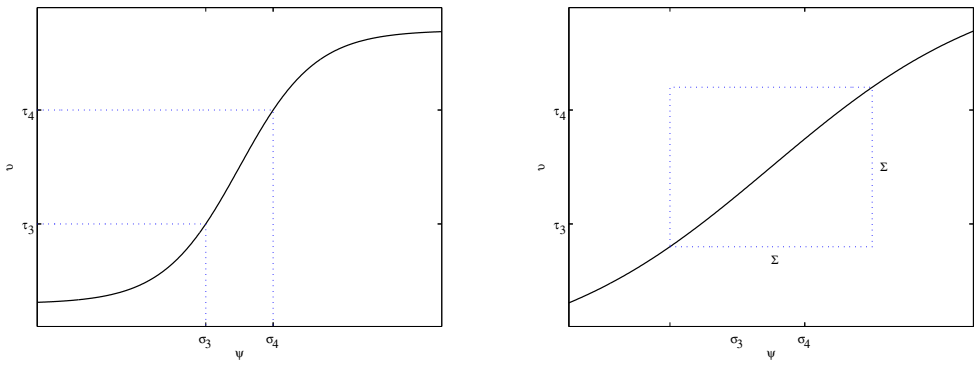


Figure 11: Graph of Heaviside Transform $u = T(\psi)$

We may use the parameter a_2 to control the value of ψ at $u = T(\psi) = 0$. We consider two cases: the first given by $T(\psi) = a_1/2$ and the second given by $T(\psi) = \tau_1$ at $\psi = 0$ where τ_1 is the lower bound of ψ. The first option will make the graph pass through zero at the midpoint of the intensity values and the second will make all values of ψ naturally positive since the lower bound of ψ will be equal to zero. Letting $u = T(\psi)$

$$u = \frac{a_1+2a_4}{1+a_2e^{\frac{-2\psi}{a_3}}} - a_4.$$

Rearranging, we have

$$a_2 = \frac{a_1+a_4-u}{e^{\frac{-2\psi}{a_3}}(u+a_4)}$$

and so for the first case, we have

$$a_2 = \frac{a_1 + a_4 - a_1/2}{a_1/2 + a_4} = \frac{a_1/2 + a_4}{a_1/2 + a_4} = 1,$$

and for the second case, we have

$$a_2 = \frac{a_1 + a_4 - \tau_1}{\tau_1 + a_4}.$$

In application, either of these will be sufficient to recover the image with similar results. In the case of the kernel, better results are obtained with $a_2 = 1$. It is there advised therefore that $a_2 = 1$ is the appropriate value for this parameter.

In summary, once a_1 and a_4 are defined, the other quantities in the transform $T(\psi) = \frac{a_1+2a_4}{1+a_2 e^{\frac{-2\psi}{a_3}}} - a_4$ can be determined automatically assuming that $\tau_3 = a_1/4$ and $a_2 = 1$ are acceptable.

References

[1] M. S. C. Almeida and L. B. Almeida. Blind and semi-blind deblurring of natural images. *IEEE T. Image Process.*, 19(1):36–52, 2010.

[2] M. A. Bahnam and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Proc. Mag.*, 14(2):24–41, 1997.

[3] L. Bar, N. Sochen, and N. Kiryati. Semi-blind image restoration via Mumford-Shah regularization. *IEEE T. Image Process.*, 15(2):483–493, 2006.

[4] J. M. Bardsley and C. R. Vogel. A nonnegatively constrained convex programming method for image reconstruction. *SIAM J. Sci. Comput.*, 25(4):1326–1343, 2004.

[5] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization.* SIAM publications, 2001.

[6] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Probl.*, 26(2):025004, 2009.

[7] Y Biraud. A new approach for increasing the resolving power by data processing. *Astron. Astrophys.*, 1:124–127, 1969.

[8] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM J. Imaging Sci.*, 3:492–526, 2010.

[9] C. Brito-Loeza and K. Chen. Multigrid method for a modified curvature driven diffusion model for image inpainting. *J. Comput. Math.*, 26(6):856–875, 2008.

[10] C. Brito-Loeza and K. Chen. Multigrid algorithm for high order denoising. *SIAM J. Imaging Sci.*, 3(3):363–389, 2010.

[11] J. F. Cai, H. Ji, C. Liu, and Z. Shen. Framelet based blind motion deblurring from a single image. *IEEE T. Image Process.*, 21:562–572, 2012.

[12] D. Calvetti, G. Landi, L. Reichel, and F. Sgallari. Nonnegativity and iterative methods for ill-posed problems. *Inverse Probl.*, 20:1747–1758, 2004.

[13] D. Calvetti, B. Lewis, L. Reichel, and F. Sgallari. Tikhonov regularization with nonnegativity constraint. *Electron. Trans. Numer. Anal.*, 18:153–173, 2004.

[14] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9:263–340, 2010.

[15] R. H. Chan, M. Tao, and X. M. Yuan. Constrained total variational deblurring models and fast algorithms based on alternating direction method of multipliers. *SIAM J. Imaging Sci.*, 6:680–697, 2013.

[16] T. F. Chan and K. Chen. On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation. *Numer. Algorithms*, 41(4):387–411, 2005.

[17] T. F. Chan and L. A. Vese. Active contours without edges. *CAM Report, UCLA*, pages 98–53, 1998.

[18] T. F. Chan and C. K. Wong. Total variation blind deconvolution. *IEEE T. Image Process.*, 7(3):370–375, March 1998.

[19] Q. Chang, X.-C. Tai, and L. Xing. A compound algorithm of denoising using second-order and fourth-order partial differential equations. *Numer. Math. Theor. Meth. Appl.*, 2:353–376, 2009.

[20] K. Chen, E. Loli Piccolomini, and F. Zama. An automatic regularization parameter selection algorithm in the total variation model for image deblurring. *Numer. Algorithms*, pages 1–20, 2013.

[21] Y. Dong, M. Hintermüller, and M. M. Rincon-Camacho. Automated regularization parameter selection in multi-scale total variation models for image restoration. *J. Math. Imaging Vis.*, 40:83–104, 2011.

[22] C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering.* SIAM publications, 2006.

[23] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms: Part 1: Fundamentals*, volume 305 of *A Series of Comprehensive Studies in Mathematics*. Springer, 1993.

[24] Y. Huang, M. K. Ng, and Y.-W. Wen. A fast total variation minimization method for image restoration. *Multiscale Model. Simul.*, 7:774–795, 2008.

[25] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Proc. Mag.*, 13(3):43–64, May 1996.

[26] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE Signal Proc. Mag.*, 13(6):61–63, Nov 1996.

[27] R. L. Lagendijk, I. Biemond, and D. E. Boekee. Regularized iterative image restoration with ringing reduction. *IEEE T. Acoust. Speech*, 36(12):1874–1888, 1988.

[28] K. Papafitsoros and C.-B. Schönlieb. A combined first and second order variational approach for image reconstruction. *J. Math. Imaging Vis.*, 48(2):308–338, 2014.

[29] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[30] M. I. Sezan and A. M. Tekalp. Survey of recent developments in digital image restoration. *Opt. Eng.*, 29(5):393–404, May 1990.

[31] M. I. Sezan and H. J. Trussell. Prototype image constraints for set-theoretic image restoration. *IEEE T. Signal Proces.*, 39(10):2275–2285, 1991.

[32] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM SIGGRAPH 2008 Papers*, 2008.

[33] Y. Shi, Q. Chang, and J. Xu. Convergence of fixed point iteration for deblurring and denoising problem. *Appl. Math. Comput.*, 189:1178–1185, 2007.

[34] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.

[35] F. Wang. *Alternating Direction Methods for Image Recovery*. PhD thesis, Hong Kong Baptist University, 2012.

[36] W. Wang and M. K. Ng. On algorithms for automatic deblurring from a single image. *J. Comput. Math.*, 30:80–100, 2012.

[37] Y. Wen and R. H. Chan. Parameter selection for total variation based image restoration using discrepancy principle. *IEEE T. Image Process.*, 21:1770–1781, 2012.

[38] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *Int. J. Comput. Vision*, 98(2):168–186, 2012.