

**TOTAL VARIATION SEMI-BLIND DECONVOLUTION USING
SHOCK FILTERS**

By

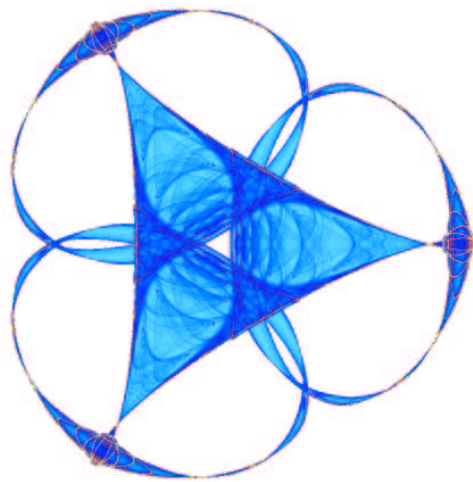
James H. Money

and

Sung Ha Kang

IMA Preprint Series # 2106

(March 2006)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455-0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: <http://www.ima.umn.edu>

Total Variation Semi-Blind Deconvolution Using Shock Filters

James H. Money and Sung Ha Kang *

March 23, 2006

Abstract

We present a Semi-Blind method for image deconvolution. This method uses a pre-processed image (via the shock filter) as an initial condition for total variation (TV) minimizing blind deconvolution. Using shock filter gives good information on location of the edges, and using variational functional such as Chan and Wong [T.F. Chan and C.K. Wong, Total variation blind deconvolution, IEEE Trans Image Process 7 (1998), 370-375] allows robust reconstructions of the image and the blur kernel. Comparison between using the L^1 and L^2 norms for the fidelity term is presented, as well as an analysis on the choice of the parameter for the kernel functional. Numerical results indicate the method is robust for both black and non-black background images while reducing the overall computational cost.

1 Introduction

We consider a problem of recovering the true image from a blurry and noisy observed image. It is also known as the image deblurring problem and has been studied for known and unknown blur kernels. In this paper, we focus on the *blind deconvolution* dealing with unknown kernels. We assume the kernel is unknown, but utilize a good initial guess to give a good clue for the correct blur kernel.

Blind deconvolution is studied by various researchers using different approaches. There are approaches using functional settings with different types of alternating minimization schemes. You and Kaveh [29] considered using an alternating scheme involving the H^1 norm for the kernel. Chan and Wong [7] extended this idea to the TV norm for both the image and the kernel, noting that in many cases the kernel function has sharp edges (such as motion blur and out of focus blur). The authors used the Alternating-Minimization [7] method

*Department of Mathematics, University of Kentucky, Lexington, KY 40515, USA, (jmoney, skang@ms.uky.edu). We acknowledge supported by grants from the NSF under contracts DMS-0312223.

for image and kernel recovery. In [18], Lin, et. al. extended the TV functional [7] to include additional constraints on the kernel and used Bregman iteration to improve the result. More recent works on spatially variant blurs and non-local functional can be found in [17, 19, 27].

Another approach for deblurring is to apply various filtering techniques. In [12], Fish et. al. considered using the Richardson-Lucy algorithm to implement an alternating minimization scheme using Bayes' Theorem, and got improved results from Weiner filter blind deconvolution. Using partial differential equation is proposed by Osher and Rudin [22] via shock filter. This method reconstructs the edges by creating shock at inflection points and finds accurate edge locations. Alvarez and Mazorra [1] considered a similar approach but pre-conditioning the image with diffusion in order to handle denoising and deblurring simultaneously. Gilboa, et. al. [15, 14] extended this idea by using a complex diffusion process.

There are considerable number of research on combining several functional in various image processing tasks. Chan, et. al. in [11] considered using blind deblurring with inpainting and calculated the solution as a single method. Bar, et. al., in [3], considered coupling with edge detection for Gaussian type kernels. In [4], the authors used the L^1 fidelity term to remove salt and pepper noise in a deblurring problem. In [2], the authors considered deblurring and impulse noise removal via a combination of the Mumford-Shah model and total variation models in a multichannel setting, and in [5], the authors combined semi-blind image restoration with segmentation for parametric blur kernels.

Given the quantity of research in the area, Chan and Shen [10, 9] present an overview of image deblurring methods developed over the past two decades, which includes stochastic methods, Tikhonov regularizations including TV regularization, and wavelet based algorithms.

In this paper, we base our model on [7] and utilize the minimizing functional,

$$\min_{k,u} \frac{1}{p} \int_{\Omega} |k \star u - u_0|^p dx dy + \lambda_1 \|k\|_{TV(\Omega)} + \lambda_2 \|u\|_{TV(\Omega)}. \quad (1)$$

Here, u is the focused image we wish to recover, k is the blurring kernel, u_0 is the observed blurry image. When $p = 2$, this is TV blind deconvolution model [7], and it is known to work well with good initial guesses. Therefore, in this paper, we introduce a good initial guess by pre-processing u_0 . We refer to this pre-processed initial guess as reference image u_r . Different pre-processing methods can be utilized, and in this paper, we apply the shock filter [22, 1] to get good edge information. We use this reference image u_r as initial condition to find the blur kernel k , then find image u separately.

In the light of applying various fitting terms such as L^1 and L^2 , we consider L^1 and L^2 norm fittings in application to this semi-blind deconvolution. In [13], Nikolova, et. al., analyzed combinations for using L^1 and L^2 fitting for image recovery with known kernels. Nikolova [21] considered using a non-convex functional with least squares method to solve the regularization problem with an emphasis on accurate edge recovery. There are various analysis for L^1 and L^2 fitting terms in denoising context [8, 16, 20] and texture decomposition in [28].

The main contribution of this paper is to drastically improve the result of using TV blind deconvolution [7] by simply utilizing the shock filter [22, 1] as initial guess, and we analyze the difference between L^1 and L^2 norm fitting term in this deblurring application. This paper is organized as follows. In Section 2, we present the details of the model as well as investigate some properties of the suggested model. In Section 3, we cover the details of numerical method as well as consider various comparisons. This is followed by concluding remarks in Section 4.

2 Semi-Blind Deconvolution Model

We consider the image model,

$$u_0 = k \star u + \eta,$$

where k is the blurring operator or kernel function, u is the true image, \star indicates convolution, and η is a noise function. We assume that $u \geq 0$, $k \geq 0$ and that $\int_{\Omega} k(x, y) dx dy = 1$.

We utilize functional (1) as in [7] and also consider the Euler-Lagrange equations approach for the kernel k and the image u . Since function is minimized for two variables, the Euler-Lagrange equations are equivalent to considering the following two functions separately,

$$\begin{aligned} \min_k \frac{1}{p} \int_{\Omega} |k \star u - u_0|^p dx dy + \lambda_1 \int_{\Omega} |\nabla k| dx dy \\ \min_u \frac{1}{p} \int_{\Omega} |k \star u - u_0|^p dx dy + \lambda_2 \int_{\Omega} |\nabla u| dx dy. \end{aligned}$$

Instead of using Alternating-Minimization (AM) method for these functionals, we separate these two functionals and use the reference image u_r as a initial condition for finding kernel k .

From the given blurry image u_0 , we apply shock filter [22] to compute the reference image u_r via solving the following PDE,

$$u_t = -|\nabla u| \text{sign}(\mathcal{L}(u)) \quad (2)$$

where $\mathcal{L}(u)$ is the second derivative in the direction of the gradient,

$$\mathcal{L}(u) = \frac{1}{|\nabla u|^2} (u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}).$$

By applying this image u_r as initial condition, we find the kernel k by minimizing the following functional,

$$\min_k \frac{1}{p} \int_{\Omega} |k \star u_r - u_0|^p dx dy + \lambda_1 \int_{\Omega} |\nabla k| dx dy. \quad (3)$$

Then, we find reconstruction u by minimizing the following,

$$\min_u \frac{1}{p} \int_{\Omega} |k \star u - u_0|^p dx dy + \lambda_2 \int_{\Omega} |\nabla u| dx dy. \quad (4)$$

We use the Euler-Lagrange equations for k and u respectively,

$$\begin{aligned} u_r(-x, -y) \star (u_r(x, y) \star k(x, y) - u_0(x, y)) - \lambda_1 \nabla \cdot \left(\frac{\nabla k}{|\nabla k|} \right) &= 0 \\ k(-x, -y) \star (k(x, y) \star u(x, y) - u_0(x, y)) - \lambda_2 \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) &= 0 \end{aligned} \quad (5)$$

Note that we solve each equation separately once, and there is no alternating process for k and u as in AM method. Since we utilize a reference image u_r for kernel reconstruction in (3), we refer to this method as *Semi-Blind deconvolution*. However, we are solving “blind” deblurring problem where we don’t assume any particular properties for the kernel nor for the image (expect that they are BV functions).

We illustrate the method in Figure 1. The reference image u_r (results of solving (2)) in (c) may not be a perfect deblurred result. However, combined with the k and u minimization equations (3) and (4), the result of Semi-Blind method in (d) recovers sharp edges and fine details. By using the shock filter, the reference image recovers sharp edges and gives good position information on these edges. With this correct information in u_r , the minimization scheme results in a better reconstruction of k which results in better image reconstruction u .

In the following subsections we consider further details of the model. We explore an analysis of the method’s error term as well as how to adaptively choose λ_1 if scale values are known (for the object in the image).

2.1 Relations between accuracy of u_r and k

In this paper, we utilized the shock filter as an initial guess to the deblurring process. In [22], Osher and Rudin introduce the idea of the shock filter as the inverse operation of diffusing a sharp image. The main idea is to reverse the effects of applying the Gaussian operator by creating a discontinuity at the point of inflection by using equation (2). However, the shock filter may not represent the true image u when the blur kernel is not Gaussian. Several researchers [1, 15] considered smoothing the image before applying the shock filter to improve the final image quality. We also apply the smoothing via the heat equation to precondition the input to the shock filter when using non-Gaussian kernels.

The shock filtered image u_r gives good clues for the edge information and the position of objects in the image as well as the shape of the kernel k . The convolution $k \star u$ is only affected when the value of u changes such as at the edges. Therefore, it is important that the edge information is correct. The location of the edges in u_r determines the relative position of the kernel k as well.

We utilized the Lagged Diffusivity Fixed Point Method (see [25, 26]) for solving Euler-Lagrange equations (5). This is a scheme where the denominator for the TV term is lagged by one step and iterated until it converges. We denote the lagged TV term by L such that

$$Lk^n = -\nabla \cdot \left(\frac{\nabla k^n}{|\nabla k^{n-1}|} \right).$$

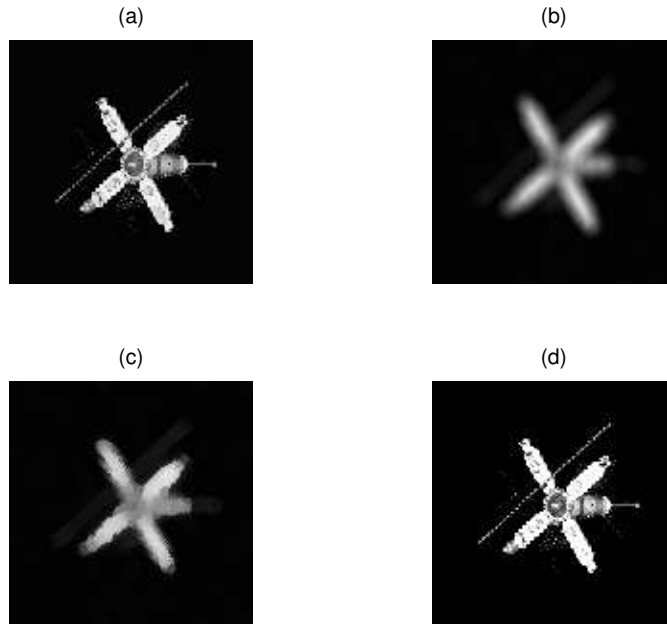


Figure 1: Illustration Semi-Blind method. (a) The true image (b) Given blurry original image u_0 (c) Reference image u_r using shock filter (d) Semi-Blind method.

The following theorem further illustrates the importance of having a good reference image u_r .

Theorem 1 *Assume the reference image is a perturbation of the true image, i.e., $u_r = u + \delta u$, then the error in the computed kernel using one step of the Lagged Diffusivity Fixed Point method is*

$$E = \delta U^T U + \delta U^T \delta U$$

where U and δU are the convolution matrix operators for u and $\delta u = u_r - u$, respectively. Hence the error bound is

$$\frac{\|\delta k\|_2}{\|k\|_2} \leq \|(U^T U - \lambda_1 L)^{-1}\|_2 (\|\delta U\|_2^2 + \|U\|_2 \|\delta U\|_2) \quad (6)$$

where $\|\cdot\|_2$ denotes the vector two norm.

Proof. The Lagged Diffusivity Fixed Point Method solves

$$U^T(Uk - u_0) - \lambda_1 Lk = 0$$

which is

$$(U^T U - \lambda_1 L) k = U^T u_0.$$

If $A = U^T U - \lambda_1 L$ and $b = U^T u_0$, we get $Ak = b$. In fact, we solve the perturbed system

$$((U + \delta U)^T (U + \delta U) - \lambda_1 L) k = (U + \delta U)^T u_0$$

or

$$(U^T U - \lambda_1 L + \delta U^T (U + \delta U)) k = U^T u_0 + \delta U^T u_0$$

Thus, $\delta A = \delta U^T (U + \delta U)$ and $\delta b = \delta U^T u_0$, where $(A + \delta A)k = b + \delta b$. By rearranging the terms, we get $A(k + \delta k) = b + \delta b$ where $A\delta k = \delta Ak$, hence $\delta k = A^{-1} \delta Ak$. Therefore,

$$\begin{aligned} \|\delta k\|_2 &\leq \|A^{-1}\|_2 \|\delta A\|_2 \|k\|_2 \\ &= \|(U^T U - \lambda_1 L)^{-1}\|_2 \|\delta U^T (U + \delta U)\|_2 \|k\|_2 \end{aligned}$$

Thus, the relative error is bounded by equation (6). \square

Note that the error of the kernel is proportional to the error in approximating the true image u by u_r . If the approximation error for u_r is large, the kernel might contain large amounts of error. In addition, the condition number of $\|(U^T U - \lambda_1 L)^{-1}\|_2$ can also affect the error; therefore, λ_1 should be chosen properly. A badly chosen λ_1 will ill-condition the matrix $U^T U - \lambda_1 L$ resulting in poor results for the kernel. Since u_r is calculated via the shock filter, the edge information and position is close to the true image, resulting in a small δU term throughout the image. As a result, the Semi-Blind method results in a small error in the computation of the kernel function.

2.2 Adaptive scale recognition for L^2

In this subsection, we consider the optimal value for λ_1 for the kernel functional (3) given the scale of the kernel and the object in the image.

We first consider one-dimensional case where a symmetric kernel k centered at the origin has a radius r over the interval $(0, 1)$ with a norm one. We assume there is one object in the image u_0 which has radius $\tilde{x} \in (0, 1)$. Let u and normalized k be given by

$$u(x) = \begin{cases} 1, & -\tilde{x} \leq x \leq \tilde{x} \\ 0, & x < -\tilde{x} \text{ or } x > \tilde{x} \end{cases} \quad \text{and} \quad k(x) = \begin{cases} 0, & x > r \text{ or } x < -r \\ \frac{1}{2r}, & -r \leq x \leq r \end{cases}.$$

We assume natural boundary conditions, i.e. the boundary value γ is extended as far as needed for the convolution operator. Between the kernel radius r and the image radius \tilde{x} , we assume that $\tilde{x} \geq r$ and $\tilde{x} - 1 \leq -\tilde{x} - r$. Figure 2 illustrates the details this setting. Assuming there is no noise, the given blurry image u_0 is defined as $u_0(x) = k * u(x) = \int_{-1}^1 k(y) u(x - y) dy$. By minimizing TV functional (3) for the kernel (with $p = 2$), the height of the kernel reduces while raising the base of the kernel to preserve the normality of the kernel (see [24, 23] for details on denoising case). In Figure 2 (a), we denote the shifts in the height of

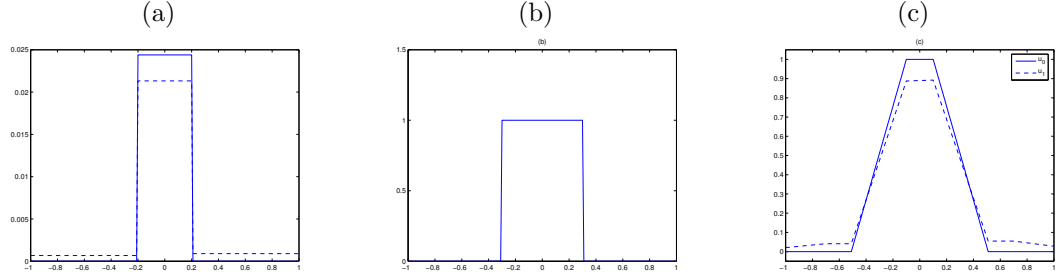


Figure 2: (a) The input kernel k (solid) and the output kernel \tilde{k} (dotted) after applying equation (3) with a sufficiently large λ_1 . (b) The true image u . (c) The given blurred image $u_0 = k \star u$ (solid) and the reconstructed image $u_1 = \tilde{k} \star u$ (dotted) with \tilde{k} . We compute the error between u_0 and u_1

the kernel function as δ_1 , δ_2 , and δ_3 , from left to right. We define the kernel \tilde{k} to be the result of applying (3),

$$\tilde{k}(x) = \begin{cases} \delta_1, & x < -r \\ \frac{1}{2r} - \delta_2, & -r \leq x \leq r \\ \delta_3, & x > r \end{cases} . \quad (7)$$

We assume that these δ_i s are chosen to normalize \tilde{k} . The following Theorem shows the relations between δ_i s and λ_1 .

Theorem 2 *The (one-dimensional) solution \tilde{k} to*

$$\min_k \left\{ \frac{1}{2} \|k \star u - u_0\|_{L^2(-1,1)}^2 + \lambda_1 \int_{-1}^1 |\nabla k| dx \right\}$$

is given by

$$\delta_1 = \delta_3 = \frac{9\lambda_1}{rp(\tilde{x}, r)}, \quad \delta_2 = \frac{3q(\tilde{x}, r)\lambda_1}{4r^2p(\tilde{x}, r)} \quad (8)$$

where

$$p(x, r) = -12r - 51x^2 + 36x + 5xr, \quad q(x, r) = 12r + 5x - 12$$

and \tilde{k} is given from (7). The parameter λ_1 is assumed to be chosen sufficiently large enough so that there is no shift in the kernel boundary regions.

Proof. The detailed notations as well as Lemmas and Theorem used to prove this Theorem can be further found at Appendix of this paper.

The solution \tilde{k} is a minimizer. Therefore, we find δ_i s which minimizes this functional by considering the derivative of the functional in terms of each δ_i s. The second term, $\lambda_1 \int_{\Omega} |\nabla \tilde{k}| dx$ of the functional, can be represented as

$$\lambda_1 \left[\left(\frac{1}{2r} - \delta_1 - \delta_2 \right) H(x+r) + \left(\frac{1}{2r} - \delta_2 - \delta_3 \right) H(x-r) \right]$$

where H is the Heaviside function. The first term $f = \|k \star u - u_0\|_{L^2(-1,1)}$ is given from Theorem 4 in Appendix. Then, we solve

$$\begin{cases} \frac{df}{d\delta_1}(\delta_1, \delta_2, \delta_3) - \lambda_1 & = 0 \\ \frac{df}{d\delta_2}(\delta_1, \delta_2, \delta_3) - 2\lambda_1 & = 0 \\ \frac{df}{d\delta_3}(\delta_1, \delta_2, \delta_3) - \lambda_1 & = 0 \end{cases}$$

to get the equation for each δ_i . \square

From this equation, notice that there is a dependence on the square of the radius of the kernel and the radius of the object in the image. This result reinforces the logic of λ_1 being proportional to the size of the kernel support as was deduced in [7]. In addition, since both image and kernel are symmetrical, the result is also symmetrical.

2.3 Semi-Blind method with L^1 norm fitting ($p = 1$)

In this section, we consider using L^1 norm fitting for image deblurring in the functionals (3) and (4). Using the L^2 norm for the fitting term classically leads to minor drop in contrast values for the resulting image and stair-casing effects. The L^1 norm typically does not suffer from this artifact in TV regularization and generally results in a dependence on the scales of objects in the image. Using $p = 1$ in equations (3) and (4), we obtain

$$\begin{aligned} \min_k \int_{\Omega} |k \star u_r - u_0| dx dy + \lambda_1 \int_{\Omega} |\nabla k| dx dy \\ \min_u \int_{\Omega} |k \star u - u_0| dx dy + \lambda_2 \int_{\Omega} |\nabla u| dx dy, \end{aligned}$$

and the corresponding Euler-Lagrange equations become

$$\begin{aligned} k(-x, -y) \star \frac{k(x, y) \star u - u_0}{|k(x, y) \star u - u_0|} - \lambda_1 \nabla \cdot \left(\frac{\nabla k}{|\nabla k|} \right) &= 0 \\ u(-x, -y) \star \frac{k(x, y) \star u - u_0}{|k(x, y) \star u - u_0|} - \lambda_2 \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) &= 0 \end{aligned} .$$

Analysis for the λ_1 in the L^1 case can be completed as in L^2 case (8). We use the same setting as in Figure 2 in Subsection 2.2. The integration of $\|k \star u - u_0\|_{L^1(-1,1)}$ can be found in the Appendix, and by using similar proof as in Theorem 2, we get the following result for λ_1 for L^1 norm fitting.

Theorem 3 *The solution \tilde{k} to*

$$\min_k \left\{ \frac{1}{2} \|k \star u - u_0\|_{L^1(-1,1)} + \lambda_1 \int_{-1}^1 |\nabla k| dx \right\}$$

is given by

$$\delta_1 = \delta_3, \quad \delta_2 = -\frac{\delta_3(8r - 7\tilde{x} - 4)}{(8r + 9\tilde{x} - 4)}$$

where \tilde{k} is defined from equation (7). In addition, λ_1 is given by

$$\lambda_1 = -\frac{1}{4} - \frac{81}{64}\tilde{x}^2 + \frac{9}{8}\tilde{x} - r^2 - \frac{1}{4}r\tilde{x} + r.$$

The parameter λ_1 is assumed to be chosen sufficiently large enough so there is no shift in the kernel boundary regions.

The symmetry is preserved for δ_1 and δ_3 as before in L^2 case. However, there is no dependence on the shift of the intensity value nor on the size of kernel, except for δ_2 . In addition, the parameter λ_1 depends solely on the size of the kernel and the size of the object in the image. As a result, we can directly determine λ_1 with the knowledge of the image object size and the support size of the kernel, and get an adaptive form for determining the kernel as in Subsection 2.2.

3 Numerical Details and Examples

There are a wide range of models and algorithms for numerical computation for image deblurring problems [6, 25, 26]. In [25], various methods for recovery of a blurry image are discussed, including a chapter dedicated to the TV based recovery of the image using fixed point methods. Vogel and Oman [26] present a more detailed analysis for the denoising case for the methods comparing the various fixed point schemes. In [6], the authors present a comparison of the fixed point, time marching, and primal-dual based methods.

In computation of the reference image u_r , we first use heat equation to smooth the image (implemented by centered differences). We then applied the shock filter, as implemented in [22],

$$u_{ij}^{n+1} = u_{ij}^n - \frac{\Delta t}{h} \sqrt{\frac{((\Delta_+^x u_{ij}^n)^+)^2 + ((\Delta_-^x u_{ij}^n)^-)^2 + ((\Delta_+^y u_{ij}^n)^+)^2 + ((\Delta_-^y u_{ij}^n)^-)^2}{((\Delta_+^x u_{ij}^n)^-)^2 + ((\Delta_-^x u_{ij}^n)^+)^2 + ((\Delta_+^y u_{ij}^n)^-)^2 + ((\Delta_-^y u_{ij}^n)^+)^2}} \text{sign}^+(L(u^n)) - \frac{\Delta t}{h} \sqrt{\frac{((\Delta_+^x u_{ij}^n)^-)^2 + ((\Delta_-^x u_{ij}^n)^+)^2 + ((\Delta_+^y u_{ij}^n)^-)^2 + ((\Delta_-^y u_{ij}^n)^+)^2}{((\Delta_+^x u_{ij}^n)^+)^2 + ((\Delta_-^x u_{ij}^n)^-)^2 + ((\Delta_+^y u_{ij}^n)^+)^2 + ((\Delta_-^y u_{ij}^n)^-)^2}} \text{sign}^-(L(u^n))$$

where $\Delta_+^x u_{ij} = u_{i+1,j} - u_{ij}$ and $\Delta_-^x u_{ij} = u_{ij} - u_{i-1,j}$ and similarly for Δ_\pm^y . The superscripts mean $F^+ = \max(0, F)$ and $F^- = \min(0, F)$. For $L(u^n) = u_{xx}u_x^2 + 2u_{xy}u_xu_y + u_{yy}u_y^2$, the centered difference schemes are used for the second order derivatives and the minmod scheme for the first order derivatives, $\text{minmod}(a, b) = \frac{\text{sign}(a) + \text{sign}(b)}{2} \min(|a|, |b|)$.

For the computation of equations (3) and (4) for $p = 2$, we utilized the Lagged Diffusivity Fixed Point method [25, 26],

$$u_r(-x, -y) \star (u_r \star k^{n+1} - u_0) - \lambda_1 \nabla \cdot \left(\frac{\nabla k^{n+1}}{|\nabla k^n|} \right) = 0$$

and similarly for (4). We utilized the FFT to compute the convolutions and used the conjugate gradient method to solve the corresponding linear systems

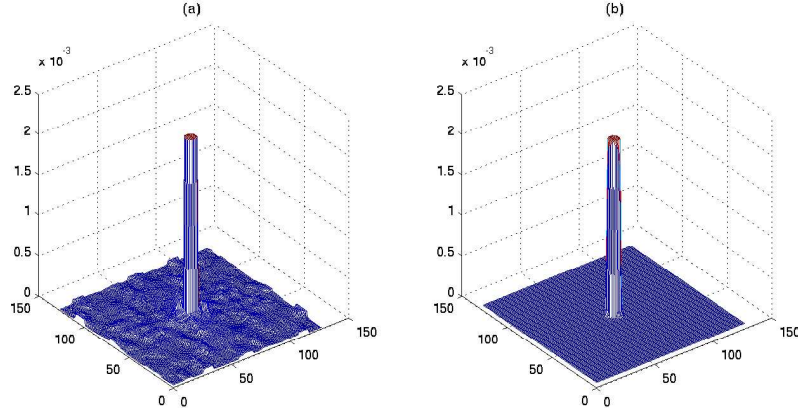


Figure 3: (a) Apparent noise levels in reconstructed kernel. (b) Adaptively denoised kernel.

at each step. In the case of $p = 1$, we use a modified fixed point method of the form

$$u_r(-x, -y) \star \left(\frac{u_r \star k^{n+1} - u_0}{|u_r \star k^n - u_0|} \right) - \lambda_1 \nabla \cdot \left(\frac{\nabla k^{n+1}}{|\nabla k^n|} \right) = 0.$$

3.1 Refinement of kernel k

When calculating the blur kernel k , there is often noise in the solution, as shown in Figure 3. Image (a) shows the calculated kernel k which has oscillation. Since calculating minimizer for kernel and image are separated process, we apply various techniques to further refine k before it is used to reconstruct image u .

From equation (8) one can estimate λ_1 and add further constraints to the model, we estimate the size of the kernel by thresholding. From the calculated k , we compute the relative size of the kernel to compute support area of k . When one uses mean value of the kernel to threshold, it generally results in a cutoff value that is too small (since the support for k is small with respect to the size of the image). Therefore, we use the threshold value to be

$$\text{cutoff} = \frac{\min(k) + \max(k)}{2}.$$

As in [24], the scale of the object is determined by $\text{scale} = \frac{|\partial D|}{|D|}$ where D is the object. For the length of the boundary, we find locations of large gradient changes. With this information, we apply an adaptive scale method for denoising (see [24, 23]) to the kernel k . Through these processes, we refine k as shown in figure 3(b).

Algorithm

1. Input u_0 , and compute u_r
2. Find the kernel by solving

$$u_r(-x, -y) \star \frac{u_r(x, y) \star k^{(1)}(x, y) - u_0(x, y)}{|u_r(x, y) \star k^{(1)}(x, y) - u_0(x, y)|^\gamma} - \lambda_1 \nabla \cdot \left(\frac{\nabla k^{(1)}}{|\nabla k^{(1)}|} \right) = 0$$
 for $k^{(1)}$ where $\gamma = 0$ for $p = 2$ and $\gamma = 1$ for $p = 1$.
 - (a) Use shock filter on $k^{(1)}$ to get $k^{(2)}$
 - (b) Normalize the kernel $k^{(2)}$.
 - (c) Denoise the kernel with small λ to get $k^{(3)}$.
 - (d) Compute $\delta_{ij} = k^{(2)} - k^{(3)}$, and threshold the image to get the scale of the kernel.
 - (e) Set $\lambda_{ij} = scale \star \delta_{ij}$, and denoise with λ_{ij} to get $k^{(4)}$.
3. Find the image by solving

$$k^{(4)}(-x, -y) \star \frac{k^{(4)}(x, y) \star u(x, y) - u_0(x, y)}{|k^{(4)}(x, y) \star u(x, y) - u_0(x, y)|^\gamma} - \lambda_2 \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0$$
 for u .

In Figure 4, we present a comparison between reconstructed images using Semi-Blind method with or without kernel refinements. Image (c) is the result without kernel refinement, and image (d) shows the reconstruction with kernel refinement. These results are compared to TV blind deconvolution using AM method, which is shown in image (b). It shows a compatible results between [7] and Semi-Blind method without refinement on k in image (c), while there is an improvement in Semi-Blind method, since more of the antenna is visible in image (c). Since the two functionals (3) and (4) are separated, image (c) can be further refined to image (d). Image (d) shows dramatic improvement in the result, and it is very close to true image.

3.2 Numerical experiments

For all of the following experiment, we used out of focus blurs for the kernel. All the images are grey scale, and sized 127 x 127. The computations were performed on a 2.0 Ghz Pentium 4 machine using Matlab.

Our first example is presented in Section 2. Figure 1 shows the deblurring result with a black background and exhibits good results with the kernel improvements when using the L^2 norm for both u and k .

Figure 5 shows an experiment with non-black background. Since the background is not black, when computing the kernel, errors in the approximation compounds instead of disappearing. This is shown in Figure 5(c) where many wrinkles are presented in the result from using [7]. Image (d) shows Semi-Blind results with refinements, and it suffers much less from this effect. The recovered image (d) is close to the true image, and even recovers the reflection on the cup. The light shadows of the cup in the true image are emphasized in the result

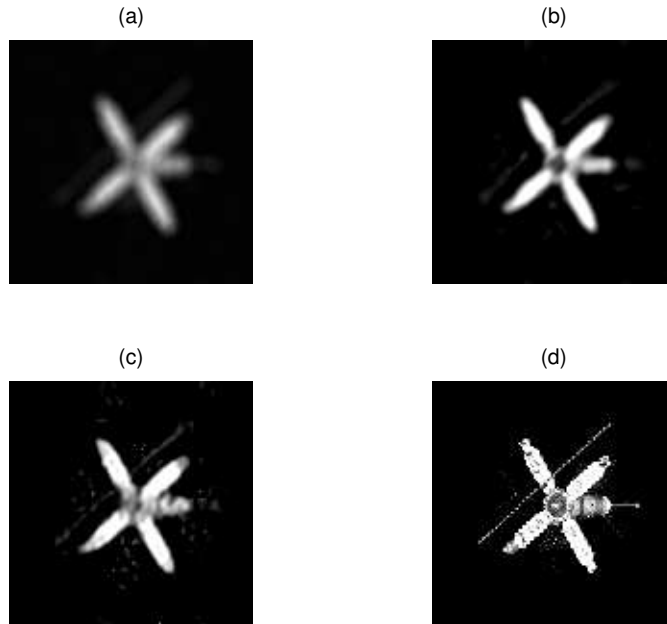


Figure 4: Comparison between AM method [7], Semi-Blind methods with or without kernel refinement. (a) The given blurry original image u_0 . (b) Chan and Wong's AM method [7] (c) Semi-Blind result without kernel refinements. (d) Semi-Blind result with kernel refinements. Image (d) shows dramatic improvement in the reconstructed image u .

(d), and there is a contrast shift. Nevertheless, it recovers the sharp edges with details from given original image (b).

This Semi-Blind model can be applied to blurry and noisy image. Figure 6 shows an example when the noise is added after blurring the image. In the result image (d), some details are lost due to denoising effect of choosing large λ_2 in (4). However, recovers reasonable image from image (b).

Since we added one step of computing the reference image u_r , we compare the computational time of the Semi-Blind method with AM method in [7]. In Figure 7, it shows the savings in computational time by only performing one iterations of kernel and image functional separately. The computational cost of the shock filter is negligible compared to TV functional iterations, and kernel refinements take only a fraction of the computational time. The overall cost of computing the Semi-Blind method with the kernel refinements is roughly less than one-third that of AM method [7] with three loop iteration.

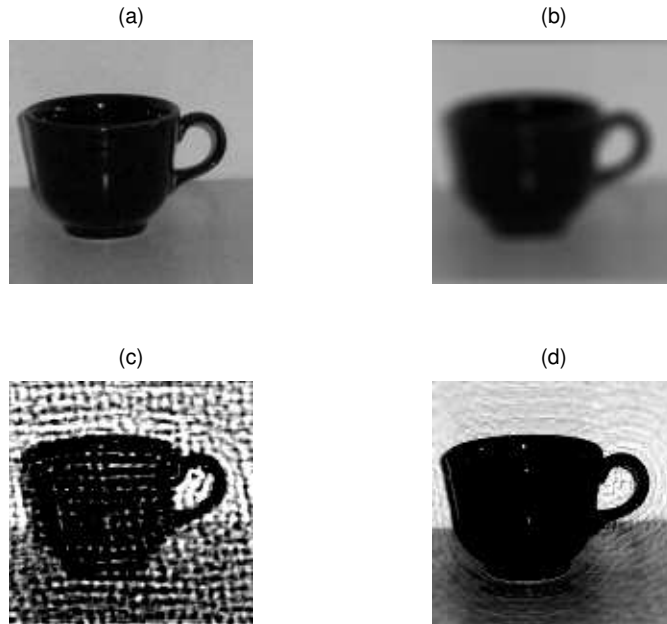


Figure 5: Non-black background image with L^2 norm fitting. (a) The true image. (b) The given blurry original image. (c) Chan and Wong [7] result. (d) Semi-Blind result (with kernel refinement). Image (d) recovers the sharp edges with details such as the reflections on the cup.

The following example illustrates using the adaptive scale selection for λ_1 in equation (3) for the kernel with $p = 2$. Figure 8 shows that adaptive scale selection gives the same result as manually choosing the optimal parameter. We applied the one dimensional analysis and used the known radius of (the out of focus) blur and used \tilde{x} to be the average radius of the object (which is the size of the cup in the picture). We see that one dimensional analysis can be accurately extended to two dimensional images.

3.3 L^1 and L^2 norm fitting comparison

In this subsection, we consider various experiments using L^1 or L^2 norm fittings for functionals (3) and (4).

First, we experiment with the original AM method [7] with different combi-

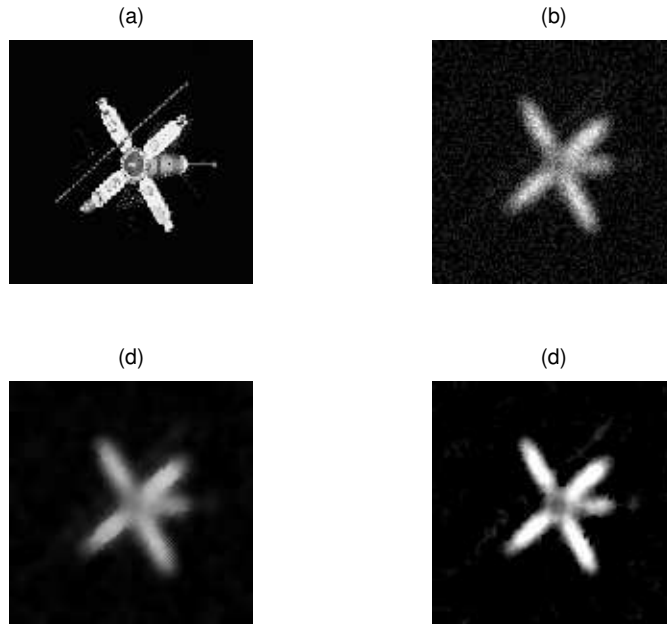


Figure 6: (a) The true image. (b) The given blurry and noisy image u_0 . (c) Shock filtered image u_r . (d) Semi-Blind result shows reconstruction u .

nations of p for two coupled equations.

$$\begin{aligned} & \min_k \frac{1}{p_1} \int_{\Omega} |k \star u - u_0|^{p_1} dx dy + \lambda_1 \int_{\Omega} |\nabla k| dx dy \\ & \min_u \frac{1}{p_2} \int_{\Omega} |k \star u - u_0|^{p_2} dx dy + \lambda_2 \int_{\Omega} |\nabla u| dx dy \end{aligned} \quad (9)$$

Figure 9 shows results of using different combinations of p_i values, either 1 or 2. Note that for this experiment, we choose λ_i to emphasize the effect of details while sacrificing smoothness of the final image to better illustrate the effects of using different p_i s. Comparing p_2 values, which is the fitting terms for the image functional in equation (9), the first column images with L^1 fitting ($p_2 = 1$) gives better details with less noisy artifacts compared to the second column images with L^2 fitting. This result is consistent with the fact that in general L^1 fitting is better for recovering details of images. Comparing different p_1 values for the kernel functional, the second row using $p_1 = 2$ gives better results in the final image. This choice will depend on the kind of true kernel used in generating u_0 . Here we used out of focus blur for the experiments, and L^2 fitting seems to recover kernel much closer to true kernel function. In this case, we conclude

Blind Deconvolution	Step	Time(secs)
Loop 1	Find kernel	299.71
	Find image	306.72
Loop 2	Find kernel	301.88
	Find image	303.63
Loop 3	Find kernel	312.71
	Find image	306.14
	Total:	1830.79

Semi-Blind Method	Time(secs.)
Initial Shock filter	0.9
Find kernel	300.19
shock kernel	2.19
adaptively denoise kernel	10.57
Find image	181.76
Total:	495.61

Figure 7: Comparison of computation time for blind deconvolution [7] and Semi-Blind method with kernel refinements.

that the best choice is image (c), using $p_1 = 2$ and $p_2 = 1$ for AM method: L^2 fitting for kernel functional and L^1 fitting for image functional.

We apply the same comparison for Semi-Blind method in Figure 10. When we compare the columns, we see that $p_2 = 1$ (L^1 fitting for image functional (4)), the first column, produces better results. By comparing the rows, we see that the second row (L^2 norm fitting for kernel functional, $p_1 = 2$) results in better images. This is similar to analysis in AM method settings. In [13], the authors presented a similar analysis using a known kernel function, which is assumed to be Gaussian. Their results show that the L^1 norm fitting is particularly well suited for images and is not affected by outlying data. This is consistent with our results for both the blind deconvolution AM method [7]) as well as Semi-Blind methods for the image functional (4).

4 Conclusion

We developed a Semi-Blind method that uses an approximation to the true image to give a good initial condition for kernel functional (3). We considered using L^1 as well as L^2 norm fittings for kernel and image functionals and found using L^1 norm fitting for image results in the best result, while L^2 was better for piecewise constant kernel functions. Adaptive parameter calculations for λ_1 is shown in Subsection 2.2 for both L^1 and L^2 norm fittings. The results show that the Semi-Blind method is robust in comparison to the AM method [7], and with the added kernel refinements, we can achieve finer details that were not normally possible with a blind deconvolution problem.

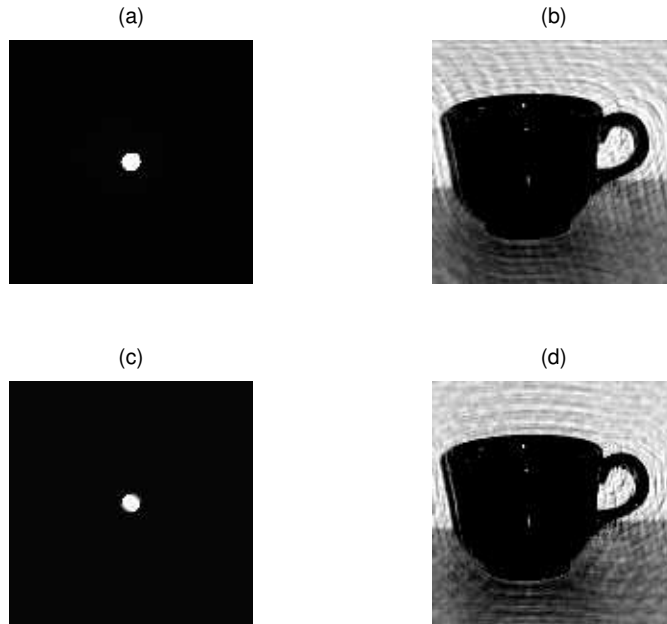


Figure 8: Adaptive scale parameter selection versus manual selection of λ_1 . In image (a) and (b), λ_1 is manually chosen, while in image (c) and (d), λ_1 is chosen adaptively. Notice that the results for the adaptive λ_1 are as accurate as manual selection of λ_1 .

Appendix

We derive the some equations used to prove theorems 2 and 3. We use the definitions from Section 2.2, and consider the case in figure 2. By labeling the jumps in the kernel function δ_1 , δ_2 , and δ_3 , we analyze the error in the approximation of the kernel when TV regularization has been applied via functional (3).

As in figure 2, the convolved functions u_0 and $u_1 = \tilde{k} \star u$ have constant slope and it can be shown that the slope depends on the constant pieces of the kernel by considering the derivative of the convolution operation of k . This lemma will assist us in analyzing the perturbed image $u_1 = \tilde{k} \star u$.

Lemma 1 For any kernel K we have that

$$\frac{d}{dx}(K \star u)(x) = \begin{cases} K(x + \tilde{x}) & x < \tilde{x} - 1 \\ K(x + \tilde{x}) - K(x - \tilde{x}) & \tilde{x} - 1 \leq x \leq 1 - \tilde{x} \\ -K(x - \tilde{x}) & x > 1 - \tilde{x} \end{cases}$$

Proof. We start by writing

$$\frac{d}{dx}(K \star u(x)) = \frac{d}{dx} \left(\int_{-1}^1 K(y)u(x-y) dy \right) = \int_{-1}^1 K(y) \frac{d}{dx} (u(x-y)) dy.$$

Note that we assumed image u has two jumps one at $-\tilde{x}$ and another at \tilde{x} . If $x < -1 + \tilde{x}$, the derivative is only affected by the left-hand side jump. Similarly, the derivative is only affected by the right jump if $x > 1 - \tilde{x}$. This results in the derivative:

$$= \int_{-1}^1 K(y) \begin{cases} \delta(x-y+\tilde{x}) & x < \tilde{x} - 1 \\ \delta(x-y+\tilde{x}) - \delta(x-y-\tilde{x}) & \tilde{x} - 1 \leq x \leq 1 - \tilde{x} \\ -\delta(x-y-\tilde{x}) & x > 1 - \tilde{x} \end{cases} dy$$

which gives the desired result. \square

This lemma allows us to assume that the pieces of the image are linear functions that can be divided into seven different parts. We use Lemma 1 to find the slope of each piece of the corresponding functions $u_0 = k \star u$ and $u_1 = \tilde{k} \star u$:

Corollary 1 We have that

$$\frac{d}{dx}u_0(x) = \begin{cases} 0 & x < -\tilde{x} - r \\ \frac{1}{2r} & -\tilde{x} - r \leq x \leq -\tilde{x} + r \\ 0 & -\tilde{x} + r < x < \tilde{x} - r \\ -\frac{1}{2r} & \tilde{x} - r \leq x \leq \tilde{x} + r \\ 0 & x > \tilde{x} + r \end{cases}$$

and

$$\frac{d}{dx}u_1(x) = \begin{cases} \delta_1 & x < -1 + \tilde{x} \\ 0 & -1 + \tilde{x} \leq x \leq -\tilde{x} - r \\ \frac{1}{2r} - \delta_1 - \delta_2 & -\tilde{x} - r \leq x \leq -\tilde{x} + r \\ \delta_3 - \delta_1 & -\tilde{x} + r < x < \tilde{x} - r \\ -\frac{1}{2r} + \delta_2 + \delta_3 & \tilde{x} - r \leq x \leq \tilde{x} + r \\ 0 & \tilde{x} + r < x < 1 - \tilde{x} \\ -\delta_3 & x \geq 1 - \tilde{x} \end{cases}.$$

The slopes of both functions depend on the slope of the lines that corresponds to the height of the kernel. In the regions of overlap, it becomes a simple

subtraction of the heights due to the combining effects of the different value for the function u . Since the function u_0 and u_1 are represented by piecewise linear functions, it is trivial to compute the respective parts of each function for preparation in computing the error in the fidelity term. That result is presented below by the following lemma.

Lemma 2 *We have that*

$$u_0(x) = \begin{cases} 0 & x > \tilde{x} + r \\ -\frac{1}{2r}(x - (\tilde{x} + r)) & \tilde{x} - r \leq x \leq \tilde{x} + r \\ 1 & -\tilde{x} + r \leq x \leq \tilde{x} - r \\ \frac{1}{2r}(x - (-\tilde{x} - r)) & -\tilde{x} - r \leq x \leq -\tilde{x} + r \\ 0 & x < -\tilde{x} - r \end{cases}$$

and

$$u_1(x) = \begin{cases} \delta_1(x + \tilde{x} + 1) & x < -1 + \tilde{x} \\ 2\delta_1\tilde{x} & -1 + \tilde{x} \leq x \leq -\tilde{x} - r \\ \left(\frac{1}{2r} - \delta_1 - \delta_2\right)(x + \tilde{x} + r) + 2\delta_1\tilde{x} & -\tilde{x} - r < x < -\tilde{x} + r \\ 2\delta_2\tilde{x} + 1 - 2\delta_1r - 2\delta_2r + (\delta_3 - \delta_1)(x + \tilde{x} - r) & -\tilde{x} + r \leq x \leq \tilde{x} - r \\ 1 - 2\delta_2r + 2\delta_3\tilde{x} - 2\delta_3r + \left(\delta_2 + \delta_3 - \frac{1}{2*r}\right)(x - \tilde{x} + r) & \tilde{x} - r < x < \tilde{x} + r \\ 2\delta_3\tilde{x} & \tilde{x} + r \leq x \leq 1 - \tilde{x} \\ -\delta_3(x - 1 + \tilde{x}) + 2\delta_3\tilde{x} & x > 1 - \tilde{x} \end{cases} .$$

Proof. This follows from Lemma 1 and the point-slope theorem for lines. \square

Using this notation for the lines for u_0 and u_1 , we compute the difference $\tilde{k} \star u - u_0$ to use it in our final calculations. This result is a simple consequence of the previous lemma and subtracting $\tilde{k} \star u = u_1$ and u_0 .

Lemma 3 *We have that*

$$\tilde{k} \star u - u_0 = u_1 - u_0 = \begin{cases} \delta_1(x + \tilde{x} + 1) & x < -1 + \tilde{x} \\ 2\delta_1\tilde{x} & -1 + \tilde{x} \leq x \leq -\tilde{x} - r \\ -\delta_2(x + \tilde{x} + r) - \delta_1(x + r) + \delta_1\tilde{x} & -\tilde{x} - r < x < -\tilde{x} + r \\ 2\delta_1\tilde{x} - 2\delta_1r - 2\delta_2r + (\delta_3 - \delta_1)(x + \tilde{x} - r) & -\tilde{x} + r \leq x \leq \tilde{x} - r \\ (\delta_2 + \delta_3)(x - r) + (\delta_3 - \delta_2)\tilde{x} & \tilde{x} - r < x < \tilde{x} + r \\ 2\delta_3\tilde{x} & \tilde{x} + r \leq x \leq 1 - \tilde{x} \\ -\delta_3(x - 1 - \tilde{x}) + 2\delta_3\tilde{x} & x > 1 - \tilde{x} \end{cases} .$$

For the L^2 fidelity term, since the sign of the function $\tilde{k} \star u - u_0$ is irrelevant, we square the previous result and integrate to obtain a reduced solution. We note that when we have the solution, there is no dependence on terms higher than degree 2, then the solution is easy to compute. The following theorem, illustrates the result for the L^2 fidelity term.

Theorem 4 *We have that*

$$\begin{aligned}
f(\delta_1, \delta_2, \delta_3) &= \frac{1}{2} \|\tilde{k} \star u - u_0\|_{L^2(-1,1)}^2 \\
&= \frac{7}{6} (\delta_1^2 + \delta_3^2) \tilde{x}^3 + 2(\delta_1^2 + \delta_3^2) \tilde{x}^2 (1 - 2\tilde{x} - r) \\
&\quad + \frac{4r}{3} [(\delta_2 + \delta_3)^2 r^2 + 3\delta_3^2 \tilde{x}^2 - 3\delta_3(\delta_2 + \delta_3) \tilde{x}r] \\
&\quad + \frac{4r}{3} [3\delta_1^2 \tilde{x}^2 - 3\delta_1^2 \tilde{x}r + \delta_1^2 r^2 + \delta_2^2 r^2 - 3\delta_1 \delta_2 \tilde{x}r + 2\delta_1 \delta_2 r^2] \\
&\quad + \frac{2(\tilde{x} - r)}{3} [2(\delta_1^2 + \delta_3^2)(\tilde{x} - r)^2 + 6\delta_2^2 r^2 - 6\delta_2(\delta_1 + \delta_3)r(\tilde{x} - r)] \\
&\quad + \frac{2(\tilde{x} - r)}{3} [-\delta_1 \delta_3 (\tilde{x} - r)^2 - 6\delta_1 \delta_3 \tilde{x}r + 3\delta_1 \delta_3 (\tilde{x}^2 + r^2)]
\end{aligned}$$

In our function (3), we compute the derivatives with respect to δ_i for $i = 1, 2, 3$ in order to minimize the functional.

We consider the case for the L^1 fidelity term. The computation follows similar as in Theorem 4, except taking into account the sign of the term $\tilde{k} \star u - u_0$. The two points where $\tilde{k} \star u - u_0$ changes sign are separated to get the corresponding integral form. These points occur in the intervals $[\tilde{x} - r, -\tilde{x} + r]$ and $[\tilde{x} - r, \tilde{x} + r]$ and are given by

$$x = -\frac{-\delta_1 \tilde{x} + \delta_1 r + \delta_2 \tilde{x} + \delta_2 r}{\delta_1 + \delta_2} \quad \text{and} \quad x = \frac{\delta_2 r - \delta_3 \tilde{x} + \delta_3 r + \delta_2 \tilde{x}}{\delta_2 + \delta_3}$$

respectively.

Theorem 5 *We have that*

$$\begin{aligned}
\|\tilde{k} \star u - u_0\|_{L^1(-1,1)} &= -\frac{\tilde{x}}{(\delta_2 + \delta_3)(\delta_1 + \delta_2)} (\delta_1^2 \tilde{x} \delta_2 - 4\delta_3^2 \delta_2 - 4\delta_3 \delta_2^2 - 4\delta_3^2 \delta_1 + \delta_3^2 \tilde{x} \delta_2 \\
&\quad + 9\delta_3 \tilde{x} \delta_2^2 + \delta_3^2 \tilde{x} \delta_1 - 8\delta_3^3 r - 4\delta_1 \delta_2^2 - 4\delta_1^2 \delta_3 - 4\delta_1^2 \delta_2 + 9\delta_1 \tilde{x} \delta_2^2 \\
&\quad + \delta_1^2 \tilde{x} \delta_3 + 4\delta_1^2 r \delta_2 + 4\delta_1^2 r \delta_3 - 4\delta_1 \delta_2^2 r \\
&\quad + 4\delta_3^2 r \delta_1 + 4\delta_3^2 r \delta_2 - 4\delta_2^2 r \delta_3 + 18\delta_1 \tilde{x} \delta_2 \delta_3 - 8\delta_1 \delta_2 \delta_3)
\end{aligned}$$

The derivatives with respect to δ_i can be calculated as in Theorem 4. These results can be used in sections 2.2 and 2.3 to determine pixel by pixel values for λ_1 in equation (5).

References

- [1] L. Alvarez and L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, 31(2):590–605, 1994.
- [2] Leah Bar, Alexander Brook, Nir Sochen, and Nahum Kiryati. Color image deblurring with impulsive noise. *Lecture Notes in Computer Science*, 3752:49–60, 2005.

- [3] Leah Bar, Nir Sochen, and Nahum Kiryati. Variational pairing of image segmentation and blind restoration. *Lecture Notes in Computer Science*, 3022:166–177, 2004.
- [4] Leah Bar, Nir Sochen, and Nahum Kiryati. Image deblurring in the presence of salt-and-pepper noise. *Lecture Notes in Computer Science*, 3459:107–118, 2005.
- [5] Leah Bar, Nir Sochen, and Nahum Kiryati. Semi-blind image restoration via mumford-shah regularization. *IEEE Transactions on Image Processing*, 15(2):483–493, 2006.
- [6] T. Chan, S. Esedoglu, F. Park, and A. Yip. Recent developments in total variation image restoration. In *“Handbook of Mathematical Models in Computer Vision”*, Springer Verlag, 2005.
- [7] T. F. Chan and C. K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–275, 1998.
- [8] Tony F. Chan and Selim Esedoglu. Aspects of total variation regularized l^1 function approximation. *SIAM Journal on Applied Mathematics*, 75(5):1817–1837, 2005.
- [9] Tony F. Chan and Jianhong Shen. *Image Processing and Analysis - Variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.
- [10] Tony F. Chan and Jianhong Shen. Theory and computation of variational image deblurring. *IMS Lecture Notes*, 2006.
- [11] Tony F. Chan, Andy M. Yip, and Frederick Park. Simultaneous total variation image inpainting and blind deconvolution. *International Journal of Imaging Systems and Technology*, 15(1):92–102, 2005.
- [12] D. Fish, A. Brinicombe, E. Pike, and J. Walker. Blind deconvolution by means of the richardsonlucy algorithm. *Journal of the Optical Society of America A*, 12(1):58–65, 1995.
- [13] Haoying Fu, Michael K. Ng, Mila Nikolova, and Jesse L. Barlow. Efficient minimization methods of mixed l^2 - l^1 and l^1 - l^1 norms for image restoration. *SIAM Journal on Scientific Computing*, 27(6):1881–1902, 2006.
- [14] G. Gilboa, N. Sochen, and Y. Y. Zeevi. Regularized shock filters and complex diffusion. *ECCV 2002, LVCS 2350, Springer-Verlag*, pages 399–413, 2002.
- [15] G. Gilboa, N. Sochen, and Y. Y. Zeevi. Image enhancement and denoising by complex diffusion processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1020–1036, 2004.
- [16] T. Karkkainen, K. Kunisch, and K. Majava. Denoising of smooth images using l^1 -fitting. *Computing*, 74(4):353–376, 2005.

- [17] Stefan Kindermann, Stanley Osher, and Peter W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 4(4):1091–1115, 2005.
- [18] He Lin, Antonio Marquina, and Stanley J. Osher. Blind deconvolution using TV regularization and bregman iteration. *International Journal of Imaging Systems and Technology*, 15(1):74–83, 2005.
- [19] James G. Nagy and Dianne P. O’Leary. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing*, 19(4):1063–1082, 1998.
- [20] Mila Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1):99–120, 2004.
- [21] Mila Nikolova. Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 4(3):960–991, 2005.
- [22] S. Osher and L. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal of Numerical Analysis*, 27(4):919–940, 1990.
- [23] D. Strong, P. Blomgren, and T. F. Chan. Spatially adaptive local feature-driven total variation minimizing image restoration. *Proceedings of the SPIE Annual Meeting*, 3137:222–233, 1997.
- [24] D. Strong and T. F. Chan. Relation of regularization parameter and scale in total variation based image denoising. Technical report, UCLA, CAM 96-07.
- [25] C. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.
- [26] C. Vogel and M. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.
- [27] Martin Welk, David Theis, and Joachim Weickert. Variational deblurring of images with uncertain and spatially variant blurs. *Pattern Recognition, Lecture Notes in Computer Science*, 3663:485–492, 2005.
- [28] Wotao Yin, Donald Goldfarb, and Stanley Osher. Image cartoon-texture decomposition and feature selection using the total variation regularized l^1 functional. *Variational, Geometric, and Level Set Methods in Computer Vision*, 3752:73–84, 2005.
- [29] Y. You and M. Kaveh. Anisotropic blind image restoration. *IEEE Transactions on Image Processing*, 8(3), 1999.

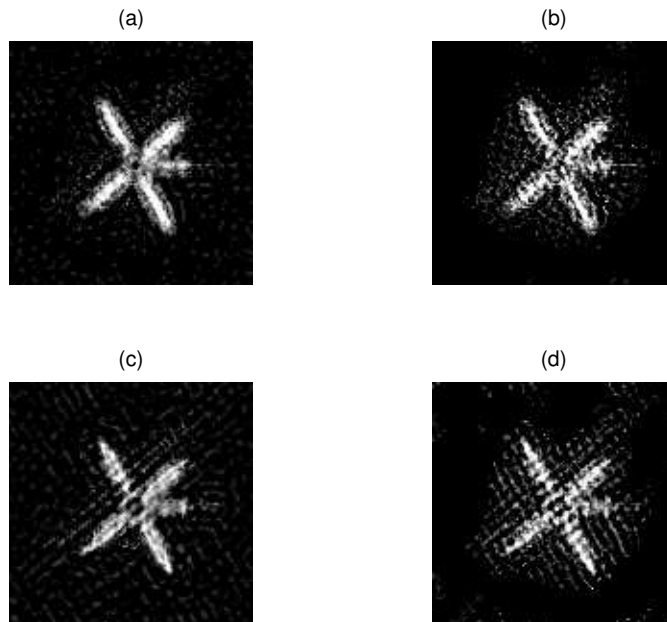


Figure 9: Comparison on using different p_1 and p_2 in AM method (1) setting. Note that we exaggerated the results by choosing λ_1 to emphasize the details in the image while sacrificing smoothness of the final image. (a) L^1 fitting for both kernel and image functional ($p_1 = p_2 = 1$). (b) L^1 fitting for kernel and L^2 fitting for image ($p_1 = 1, p_2 = 2$). (c) L^2 fitting for kernel and L^1 fitting for image ($p_1 = 2, p_2 = 1$). (d) The original AM method [7] with $p_1 = p_2 = 2$. The first column images with $p_2 = 1$ have more details recovered, and the second row images with $p_1 = 2$ has clearer results.

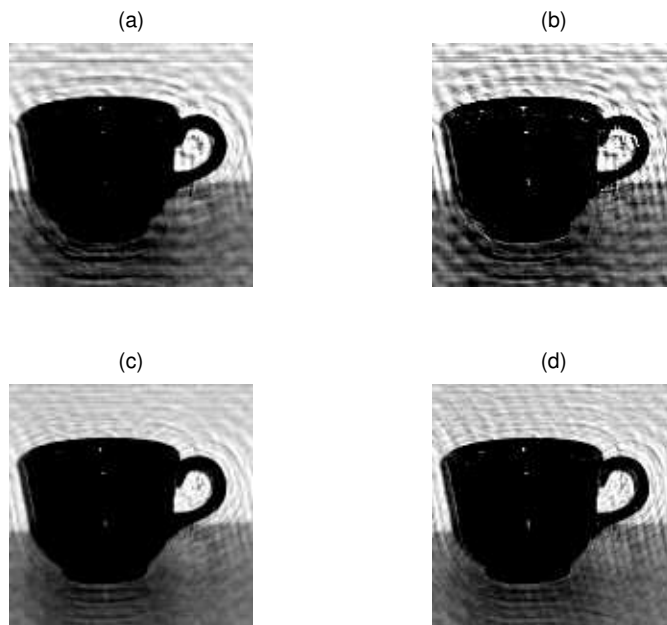


Figure 10: Comparison on using different p_1 and p_2 for Semi-Blind method. (a) L^1 fitting for both kernel and image functional ($p_1 = p_2 = 1$). (b) L^1 fitting for kernel and L^2 fitting for image ($p_1 = 1, p_2 = 2$). (c) L^2 fitting for kernel and L^1 fitting for image ($p_1 = 2, p_2 = 1$). (d) L^2 fitting for both kernel and image functional ($p_1 = p_2 = 2$). The first column images with $p_2 = 1$ have less “ringing” effects, and the second row images with $p_1 = 2$ has clearer results.