Total Variation Blind Deconvolution

Tony F. Chan and Chiu-Kwong Wong

***Abstract*—In this paper, we present a blind deconvolution algorithm based on the total variational (TV) minimization method proposed in [11]. The motivation for regularizing with the TV norm is that it is extremely effective for recovering edges of images [11] as well as some blurring functions, e.g., motion blur and out-of-focus blur. An alternating minimization (AM) implicit iterative scheme is devised to recover the image and simultaneously identify the point spread function (psf). Numerical results indicate that the iterative scheme is quite robust, converges very fast (especially for discontinuous blur), and both the image and the psf can be recovered under the presence of high noise level. Finally, we remark that psf's without sharp edges, e.g., Gaussian blur, can also be identified through the TV approach.**

***Index Terms*—Blind deconvolution, conjugate gradient method, total variation.**

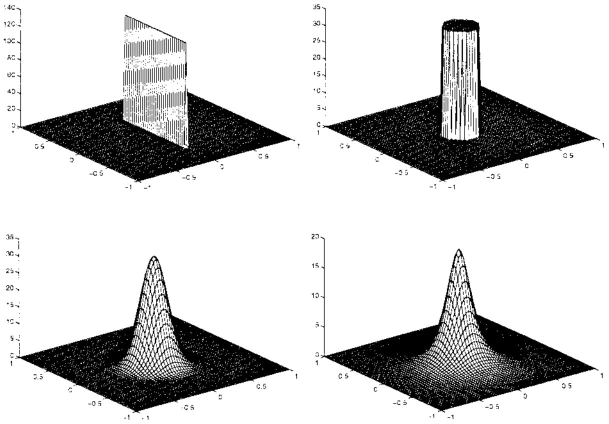

Fig. 1. Some typical psf's.

I. Introduction

It is well known that recovering the image u (resp., the point spread function, or psf, h) with known psf (resp., image) is a mathematically ill-posed problem. One of the most successful regularization approaches is the TV regularization method [11], which can effectively recover edges of an image. The mathematical formulations that we used in the image recovery problem is stated as follows [12]:

$$\min_u f(u) \equiv \min_u \tfrac{1}{2} \|h \star u - z\|^2_{L^2(\Omega)} + \alpha \int_\Omega |\nabla u| \, dx \, dy \quad (1)$$

where α is a parameter and $\star$ denotes the convolution operator. It is known that TV regularization works effectively for recovering "blocky" images [6].

In this paper, we are going to recover u and h without any *a priori* knowledge of the psf and the image. There are many existing algorithms for simultaneously identify u and h, see for instance, [7], [9], [10], [13], and [14]. As this *blind* deconvolution problem is ill-posed with respect to both h and u, You and Kaveh [13] proposed regularizing u and h by considering the following joint minimization problem:

$$\min_{u,\,h} f(u,\, h) \equiv \min_{u,\,h} \tfrac{1}{2} \|h \star u - z\|^2_{L^2(\Omega)} + \alpha_1 \|u\|^2_{H^1} + \alpha_2 \|h\|^2_{H^1}.$$

In our work, we will follow the approach in [13] and combine it with the TV regularization approach (1). More precisely, we regularize both the image and psf by the TV norm instead of the H^1 norm. The motivation for using TV regularization for the psf is due to the fact that some psf's can have edges. Fig. 1 shows the four typical psf's [8] out of which the motion blur and the out-of-focus blur are piecewise constant functions with discontinuities.

We formulate the blind deconvolution problem as

$$\min_{u,\,h} f(u,\, h) \equiv \min_{u,\,h} \tfrac{1}{2} \|h \star u - z\|^2_{L^2(\Omega)} + \alpha_1 \int_\Omega |\nabla u| \, dx \, dy + \alpha_2 \int_\Omega |\nabla h| \, dx \, dy. \quad (2)$$

Here α_1 and α_2 are positive parameters which measure the trade off between a good fit and the regularity of the solutions u and h. Such an approach of using TV as a special case of anisotropic diffusion for recovering u and h is also employed independently by You and Kaveh in their more recent work [14]. In our work, we focus on devising fast numerical algorithms for solving the minimization problem (2), which are derived from previous works on numerical methods for solving the image restoration problem (1) when h is known [2]–[4], [12]. Moreover, our algorithm can recover both the images and psf's without any *a priori* information on the psf (e.g., support size). A preliminary version of our work can be found in [5].

In the next section, we will devise an alternating minimization scheme, which is an efficient method for solving (2). Section II discusses how to choose the regularization parameters α_1 and α_2. Finally, numerical results will be presented in Section IV.

II. Blind Deconvolution by TV Regularization

To devise numerical schemes for (2), let us write down the first order optimality conditions, namely

$$\frac{\partial f}{\partial h} = u(-x,\, -y) \star (u \star h - z) - \alpha_2 \nabla \cdot \left(\frac{\nabla h}{|\nabla h|} \right) = 0, \quad x \in \Omega, \quad (3)$$

Manuscript received November 1, 1996; revised March 11, 1997. This work was supported by the Office of Naval Research under Grant ONR-N00017-96-1-0277 and by the NSF under Grant DMS-96-26755. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Guillermo Sapiro.

The authors are with the Department of Mathematics, University of California, Los Angeles, CA 90095-1555 USA (e-mail: chan@math.ucla.edu; ckwong@math.ucla.edu).

Publisher Item Identifier S 1057-7149(98)01742-4.

1057–7149/98$10.00 © 1998 IEEE

and

$$\frac{\partial f}{\partial u} = h(-x,\, -y) \star (h \star u - z) - \alpha_1 \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0, \quad x \in \Omega. \tag{4}$$

Before we solve for u and h, it is useful to note that for given u (resp., h), $f(u,\, \cdot)$ [resp., $f(\cdot,\, h)$) is convex function with respect to h (or u]. However, we remark that $f(u,\, h)$ is not jointly convex in general. Therefore, with an initial guess $(u^0,\, h^0)$ for $(u,\, h)$, we can minimize (2) by first solving $f(u^0,\, h^1) \equiv \min_h f(u^0, \cdot)$ using (3) and then $f(u^1,\, h^1) \equiv \min_u f(\cdot, h^1)$ using (4). Hence, we develop an alternating minimization (AM) algorithm in which the function value $f(u^n,\, h^n)$ always decreases as n increases. More precisely, the algorithm is stated as follows.

Assume we have u^n and h^n, proceed as follows.

- Solve for h^{n+1}

$$u^n(-x,\, -y) \star (u^n \star h^{n+1} - z) - \alpha_2 \nabla \cdot \left(\frac{\nabla h^{n+1}}{|\nabla h^{n+1}|} \right) = 0. \tag{5}$$

- Solve for u^{n+1}

$$h^{n+1}(-x,\, -y) \star (h^{n+1} \star u^{n+1} - z) - \alpha_1 \nabla \cdot \left(\frac{\nabla u^{n+1}}{|\nabla u^{n+1}|} \right) = 0. \tag{6}$$

We remark that a variant of the AM algorithm is to solve (6) first before (5) which corresponds to solve the minimization problem, $f(u^1,\, h^0) \equiv \min_u f(\cdot,\, h^0)$ first and then $f(u^1,\, h^1) \equiv \min_h f(u^1,\, \cdot)$.

There are some existing numerical methods for solving the above nonlinear type PDE's, for instance, time marching [11], lagged diffusivity fixed point (FP) schemes [12] and primal-dual methods [3]. Due to the robustness and simplicity of implementation of the fixed point algorithm, we apply it to solve (5) and (6) in this paper. The idea of the FP method is to first linearize the nonlinear PDE's (5) and (6) by lagging the diffusive coefficients $1/|\nabla h^{n+1}|$ and $1/|\nabla u^{n+1}|$ by one iteration, and then apply the FP method to solve linear problems for h^{n+1} and u^{n+1}, respectively. More precisely, the FP iterative method is described as follows.

- Solve for h^{n+1} by (iterating on i)

$$u^n(-x,\, -y) \star (u^n \star h^{n+1}_{i+1} - z) - \alpha_2 \nabla \cdot \left(\frac{\nabla h^{n+1}_{i+1}}{|\nabla h^{n+1}_i|} \right) = 0. \tag{7}$$

- Solve for u^{n+1} by (iterating on i)

$$h^{n+1}(-x,\, -y) x \star (h^{n+1} \star u^{n+1}_{i+1} - z) - \alpha_1 \nabla \cdot \left(\frac{\nabla u^{n+1}_{i+1}}{|\nabla u^{n+1}_i|} \right) = 0. \tag{8}$$

After discretizing (7) and (8), determining h_i^{n+1} and u_i^{n+1} amounts to solving two independent linear systems. It was discussed in [2] how to solve such systems by applying the conjugate gradient (CG) method, in which cosine transform preconditioners significantly speed up the convergence rate of the method. The cost per CG iteration will be dominated by two two-dimensional (2-D) fast Fourier transform (FFT) operations, which is just slightly higher than the $O(n^2)$ operating cost for "no preconditioning." However, the numerical results in [2] showed that the number of CG iterations before and after preconditioning have a great improvement, from $O(n^{0.9})$–$O(n^{0.22})$. Therefore, the overall computational cost for the CG with cosine transform preconditioner provides an efficient numerical solver for the FP scheme.

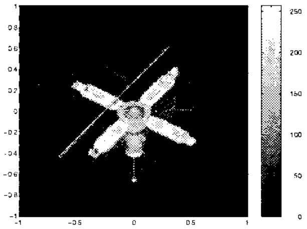

Fig. 2. Satellite image.

Unfortunately, numerical experiments indicate that the AM algorithm as stated above in its simplest form, does not always yield physical solutions. It is because the minimization problem (2) may not have a unique solution. For example, it can be easily observed that if $(u,\, h)$ is a solution, then so are $[(\alpha_2/\alpha_1)h,\, (\alpha_1/\alpha_2)u]$, $(-u,\, -h)$, $[u(x \pm c,\, y \pm d),\, h(x \mp c,\, y \mp d)]$ for any real constant c and d.

In order to obtain a physical solution, we need to impose conditions on u and h. We have chosen to impose the following conditions:

$$\int_\Omega h(x,\, y)\, dx\, dy = 1, \qquad u(x,\, y),\, h(x,\, y) \geq 0 \tag{9}$$

and h is centrosymmetric, i.e.,

$$h(x,\, y) = h(-x,\, -y).$$

We remark that besides the above conditions, we don't have any other *a priori* assumptions on the psf, e.g., the type or the support size of blurring function. We have no proof yet that (9) guarantees uniqueness of (2) but in practice (9) leads to tremendous improvement in performance, robustness and convergence of the algorithm.

In the following, we give the full details of the algorithm.

A. AM Algorithm

- Start with $u^0 = z$ and $h^0 = \delta(x,\, y)$, the delta function. Assume we have u^n and h^n,
- Solve for h^{n+1} by (iterating on i)

$$u^n(-x,\, -y) x \star (u^n \star h^{n+1}_{i+1} - z) - \alpha_2 \nabla \cdot \left(\frac{\nabla h^{n+1}_{i+1}}{|\nabla h^{n+1}_i|} \right) = 0.$$

Impose

$$h^{n+1}(x,\, y) = \begin{cases} h^{n+1}(x,\, y), & \text{if } h^{n+1}(x,\, y) > 0, \\ 0 & \text{otherwise} \end{cases}$$

$$h(x,\, y) = [h(x,\, y) + h(-x,\, -y)]/2$$

$$h^{n+1} = \frac{h^{n+1}}{\displaystyle\int_\Omega h^{n+1}(x,\, y)\, dx\, dy.}$$

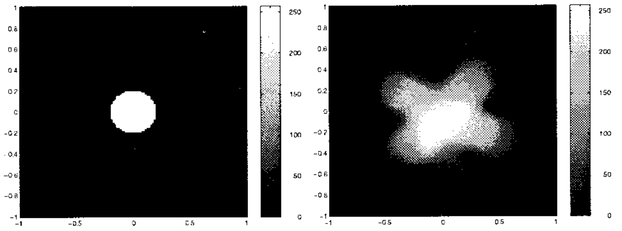

Fig. 3. Out of focus blur (left) and blurred image (right).

- Solve for u^{n+1} by (iterating on i)

$$h^{n+1}(-x,\,-y)\star(h^{n+1}\star u_{i+1}^{n+1}-z)-\alpha_1\nabla\cdot\left(\frac{\nabla u_{i+1}^{n+1}}{|\nabla u_i^{n+1}|}\right)=0.$$

Impose

$$u^{n+1}(x,\,y)=\begin{cases}u^{n+1}(x,\,y), & \text{if } u^{n+1}(x,\,y)>0\\ 0, & \text{otherwise.}\end{cases}$$

III. Choice of Regularization Parameters

In this section, we are going to point out that the regularization parameters α_1 and α_2, respectively, depends directly on the noise level and the severity of the blur in the observed image. To understand this, let us first consider the following noise-constrained minimization problem:

$$\min_{u,h}\int(|\nabla u|+\alpha|\nabla h|)\,dx\,dy \tag{10}$$

subject to

$$\|h\star u-z\|^2=\sigma^2$$

where σ is the noise level. The Lagrangian for (10) is

$$f(u,\,h)=\int|\nabla u|+\alpha|\nabla h|\,dx\,dy+\frac{\lambda}{2}(\|h\star u-z\|^2-\sigma^2)$$

where λ is the Lagrange multiplier. It follows that the minimization problems (2) and (10) are identical if $\alpha_1=2/\lambda$ and $\alpha_2=2\alpha/\lambda$. By making use of the noise-constrained formulation (10), we are going to describe some guidelines for selecting the parameters, α_1 and α_2.

Clearly, if the signal-to-noise ratio (SNR) is small (or σ is large), then λ should be small so that $\int|\nabla u|$ is sufficiently large to regularize the image. Therefore, a good heuristic is to assume that λ is proportional to the SNR. Hence, we expect $\alpha_1=2/\lambda$ to be directly proportional to the noise level σ and the numerics we have so far also support this argument (see [12]). The parameter α_2, on the other hand, controls the support or the spread of the psf. When α_2 gets bigger, the TV regularization for h, $\int|\nabla h|$, is required to be small in order to minimize (10). Therefore, the peak of h will be lower when α_2 gets bigger. Since we impose the constraint, $\int h=1$, the psf must spread out. Hence, α_2 can be chosen proportionately according to the amount of desired deblurring.

In our numerical experiments, the initial guess for u is chosen (unless stated otherwise) to be the observed image z as it is the only available approximation of u. The initial guess for h is chosen to be the delta function $\delta(x,\,y)$ because in the case of no blurring, $\delta(x,\,y)$ would be the expected psf (which can be achieved by setting $\alpha_2=0$). If the recovered image is not deblurred enough, we restart the AM algorithm with a larger value of α_2, which increases the support of the identified psf, until a reasonable recovered image appears. We will illustrate the above ideas experimentally in the next section.

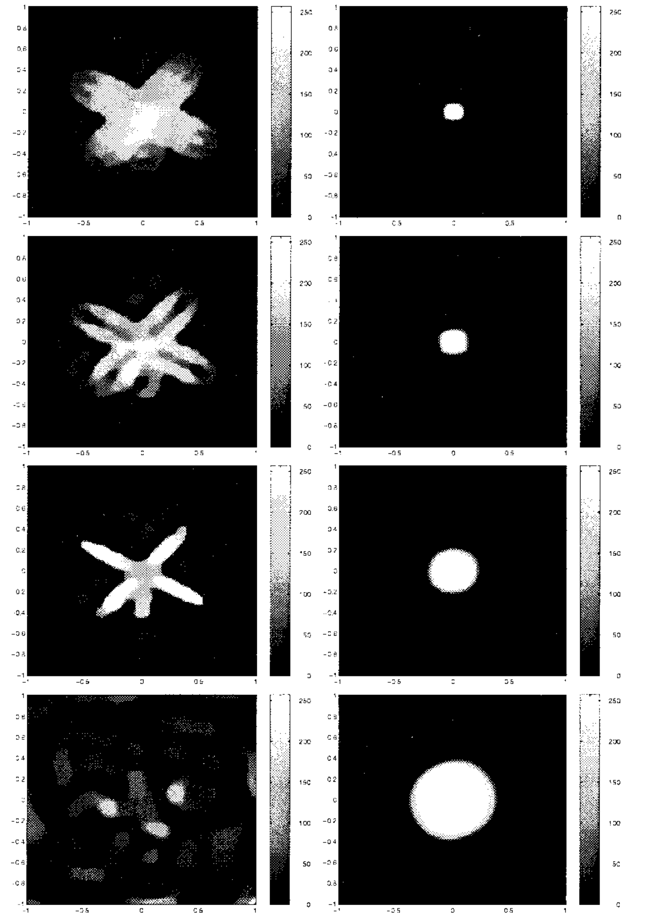

Fig. 4. Recovered images (left) and psf's (right) with $\alpha_2=10^{-7},10^{-6},10^{-5},10^{-4}$, (from top to bottom) and $\alpha_1=2\times\ 10^{-6}$. Observe that as α_2 increases, the spread of the identified psf also increases. When α_2 is too small (e.g., $\alpha_2=10^{-6}$ or 10^{-7}), the images are not sufficiently deblurred. When $\alpha=10^{-5}$, the image is in sharp focus. When we further increase α_2, the image becomes irregular.

Finally, we can combine this idea with the method of continuation on α_2. More precisely, we start the algorithm with a small α_2 and $u^0=z$ and $h^0=\delta$. When we restart the AM algorithm with a larger α_2, we can make use of the solution $(u,\,h)$ from the smaller α_2 as an initial guess, which should be a better approximation to the true solution than $(u^0,\,h^0)$. We have not explored this idea fully in this paper.

IV. Numerical Results

In this section, numerical results are presented to illustrate the efficiency and the effectiveness of the AM algorithm. The results show that the image and psf can be recovered even under the presence of high noise level with just a few AM iterations. We will also compare images that are recovered

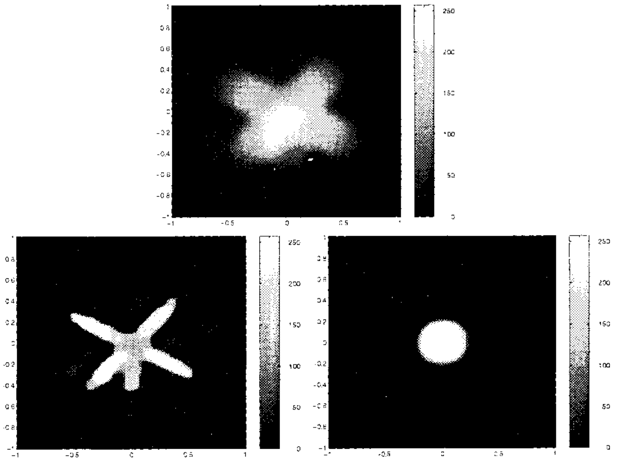

Fig. 5. Observed (top) and recovered (bottom left) image and identified psf (bottom right) after three AM iterations. $\alpha_1 = 2 \times 10^{-6}$, $\alpha_2 = 1.5 \times 10^{-5}$, SNR = ∞.

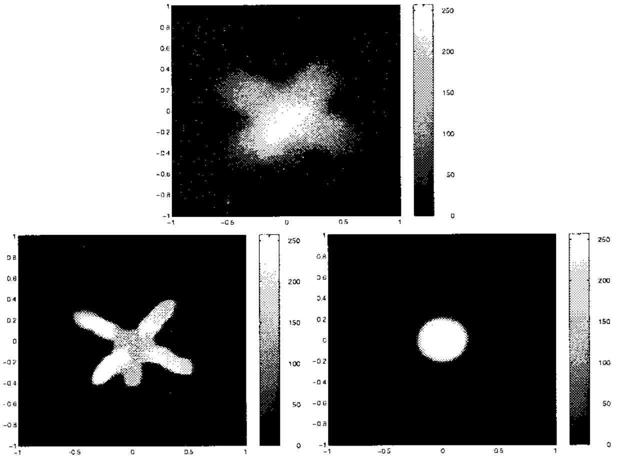

Fig. 6. Observed (top) and recovered (bottom left) image and identified psf (bottom right) after three AM iterations. $\alpha_1 = 2 \times 10^{-5}$, $\alpha_2 = 1.5 \times 10^{-5}$, SNR = 5.

by using the TV norm with those obtained by the H^1 norm. Moreover, we will perform an experiment to support the continuation idea proposed in Section III on how to choose the regularization parameter α_2.

The test image is the satellite image shown in Fig. 2 from Phillips Laboratory, Kirkland Air Force Base, NM, and was provided to us by Prof. R. Plemmons, Wake Forest University. The image originally consists of 256 × 256 pixels. To simplify the computational work, we down sample it to become a 127 × 127 pixels image. Currently, our codes are written in MATLAB with machine precision roughly equal to 10^{-16}. At each step of the AM algorithm, we iterate the FP iteration ten times. Within each FP iteration, we are required to solve a linear system and we do it by applying the CG method. In order to speed up the convergence rate of the CG iteration, we preconditioned the linear system by a cosine transform preconditioner that we developed earlier [2]. The CG iteration is stopped when the relative residual is less than 0.1. We have find that such a low tolerance for the inner iteration is good enough and is most effective for the FP method.

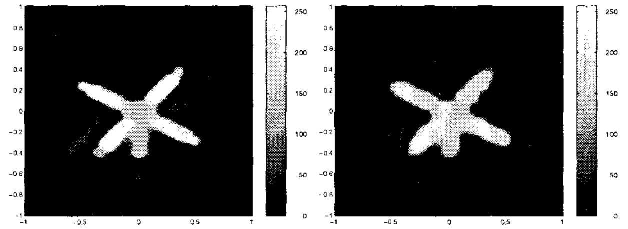

Fig. 7. Recovered image by (1) with known psf. Left: SNR = ∞, $\alpha = 2 \times 10^{-6}$, Right: SNR = 5, $\alpha = 2 \times 10^{-5}$.

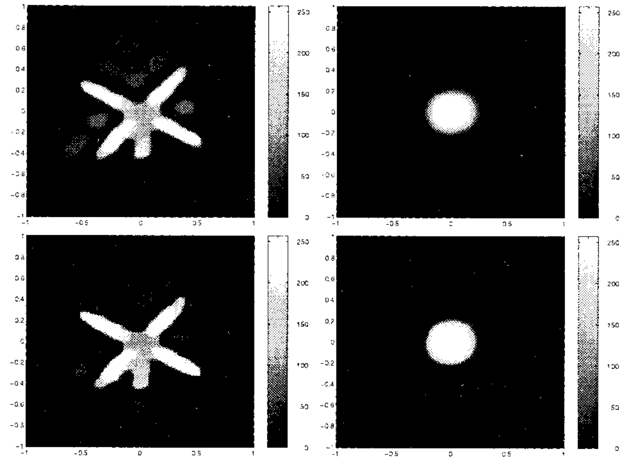

Fig. 8. Recovered images and psf's after one and three AM iterations (from top to bottom). These figures show that the AM algorithm can recover very good images after only a few iterations.

Fig. 9 shows the recovered u and h by minimizing (2) *except* we replaced the TV norm in h by the H^1 norm. In the experiment, α_1 is fixed at the value 2 × 10^{-6} and we vary α_2. We can observe that the recovered images are not as good as using the TV norm for h. This justifies why we use the TV norm when recovering h instead of the H^1 norm.

Figs. 3 and 4 illustrate the ideas on choosing the regularization parameters proposed in Section III. Fig. 3 shows the psf, which is an out-of-focus blur, and the blurred image (without noise). Fig. 4 shows the recovered images and the identified psf's for varies α_2. The parameter α_1 is fixed at 2 × 10^{-6} as it produces the best recovered image in case the psf is known. We see that the recovered psf has increasing support as α_2 increases. When $\alpha_2 = 10^{-6}$ or 10^{-7}, the support of the identified psf's are smaller than the true psf (cf., Fig. 3), and the recovered images are only slightly deblurred. When $\alpha_2 = 10^{-5}$, the recovered image is in sharp focus. If we further increase the value of α_2, says to 10^{-4}, the spread of the psf becomes too large and the recovered image becomes irregular. Therefore, by varying α_2, we can easily pick up the appropriate recovered image.

In Figs. 5–7, we show that the AM algorithm can effectively recover images and identifying psf's even in the presence of high noise level. Fig. 5 shows the observed image which is blurred by the out-of-focus blur shown in Fig. 3. The observed image in Fig. 6 is in addition polluted by Gaussian noise with SNR = 5. After performing three AM iterations on both the observed images, we obtain very good recovered images. In

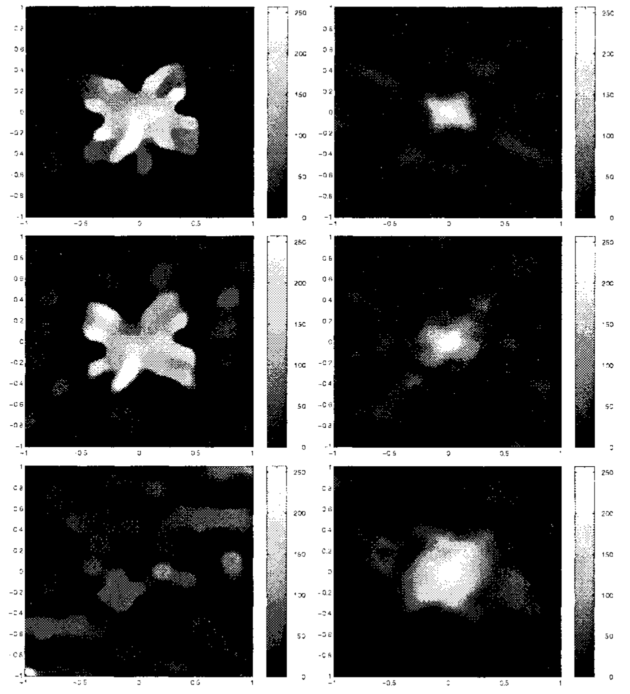

Fig. 9. Recovered image (left) and psf (right) using TV norm in u and H^1 norm in h, with $\alpha_2 = 10^{-5}$, 10^{-4}, 10^{-3} (from top to bottom). Observe that H^1 in h is not as good as TV in h.

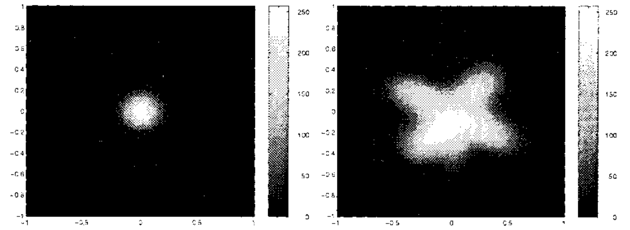

Fig. 10. Psf: Gaussian blur (left), observed image (right).

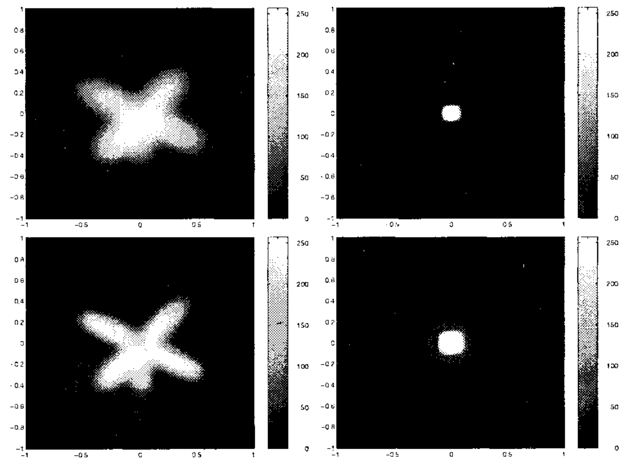

Fig. 11. Recovered image (left) and psf (right) after three (top) and 35 (bottom) AM iterations. TV norm in u and h. $\alpha_1 = 10^{-7}$, $\alpha_2 = 10^{-8}$.

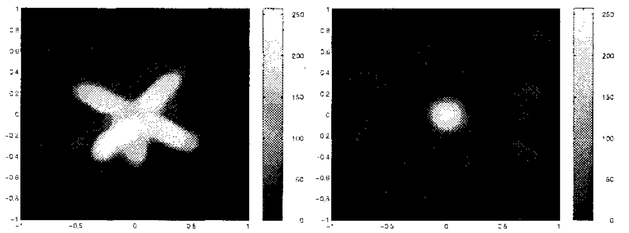

Fig. 12. Recovered image (left) and psf (right) after three AM steps. TV norm in u and H^1 norm in h. $\alpha_1 = 10^{-6}$, $\alpha_2 = 10^{-6}$.

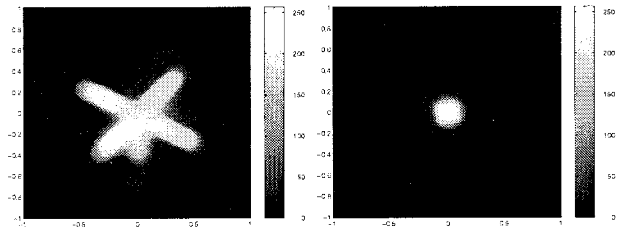

Fig. 13. Recovered images and psf's after five AM iterations with $h^0 = g_{300}$. $\alpha_1 = 10^{-7}$, $\alpha_2 = 10^{-8}$.

particular, we can see in Fig. 5 that an antenna appears in the recovered image (cf., Fig. 3). Moreover, even with a high noise level (Fig. 6), the AM algorithm can still recover a very sharp image. In Fig. 7, we display the images that are recovered from the TV regularization scheme (1) assuming the exact psf is known. Comparing the recovered images in Figs. 5–7, we find that even if we don't know the exact psf, the AM algorithm can still recover images that are almost as good as that recovered with the exact psf. This experiment demonstrates the robustness of the AM algorithm.

In Fig. 8, we display the recovered images after one and three AM iterations. We see that after the first AM iteration, we already have a very good recovered image. Hence, the AM algorithm is a efficient method for minimizing (2).

Finally, we show in Figs. 10–12 that the TV regularization approach is also good for identifying psf's without edges (e.g., the Gaussian blur, $g_\gamma(x, y) = \exp[-\gamma(x^2 + y^2)]$). Fig. 10 shows the observed image, which is blurred by Gaussian blur (with $\gamma = 200$) without noise. Unfortunately, we observe that the convergence rate of the AM algorithm is slower than in the case of out-of-focus blur. For example, Fig. 11 shows that there is still a lot of improvement in the recovered image from 3–35 AM iterations. This is almost surely due to the ill conditioning of the Gaussian blur operator. Although the psf h is not perfectly identified, we can still recover an image u, which is even better than the image obtained by solving (2) with H^1 regularization in h (cf., Fig. 12). However, we should note that the AM algorithm converges much faster when we use H^1 regularization on h. One way to speed up the convergence rate of the AM algorithm is by using a good initial guess for h (e.g., a guide star). In this case, we perform a variant of the AM algorithm, in which at each AM step, we recover u before h. Fig. 13 shows that if we make use as initial guess for h a blurring operator of the correct type (Gaussian in this case) but with a slightly wrong parameter γ ($\gamma = 300$ instead of $\gamma = 200$), then we can obtain a better solution than starting with the generic delta function (cf., Fig. 11). In fact, if our initial guess for h happens to be the exact solution g_{200} (see Fig. 14), the algorithm will leave this alone and produce almost an exact solution in three iterations.

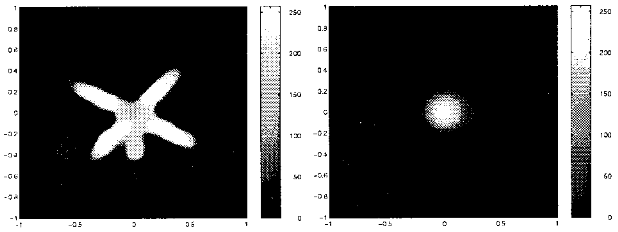

Fig. 14. Recovered images and psf's after five AM iterations with initial guess $h^0 = g_{200}$. $\alpha_1 = 10^{-7}$, $\alpha_2 = 10^{-8}$.

V. Conclusion

We conclude that the AM/FP algorithm is a robust and efficient method for solving the minimization problem (2) and we have justified the choice of using the TV norm for regularizing u and h instead of the H^1 norm. The case of highly ill-conditioned blurring operators such as Gaussian blur still requires more investigation to improve its rate of convergence.

Acknowledgment

The authors wish to thank Dr. G. Sapiro for helpful discussions and for bringing [14] to our attention.

References

[1] R. Acar and C. Vogel, "Analysis of bounded variation penalty methods," *Inv. Prob.,* vol. 10, pp. 1217–1229, 1994.

[2] R. Chan, T. Chan, and C. Wong, "Cosine transform based preconditioners for total variation deblurring," CAM Rep. 97-44, Univ. Calif., Los Angeles, CA, Sept. 1997. An earlier version appeared in *Iterative Methods in Linear Algebra, II,* vol. 3, S. Margenov and P. Vassilevski, Eds. IMACS Series in Computational and Applied Math, 1996, pp. 311–329.

[3] T. Chan, G. Golub, and P. Mulet, "A nonlinear primal-dual method for TV-based image restoration," in *Proc. 12th Int. Conf. Analysis and Optimization of Systems: Images, Wavelets and PDE's,* M. Berger *et al.,* Eds., Paris, France, June 26–28, 1996..

[4] T. Chan and P. Mulet, "Iterative methods for total variation image restoration," in *Proc. Winter School on Iterative Methods,* Chinese University of Hong Kong. New York: Springer-Verlag, 1995.

[5] T. Chan and C. Wong, "Total variation blind deconvolution," in *Proc. ONR Workshop,* Sept. 4–6, 1996, WWW: http://www.math.ucla.edu/∼blomgren/Workshop96/.

[6] D. Dobson and F. Santosa, "Recovery of blocky images for noisy and blurred data," *SIAM J. Appl. Math.,* vol. 56, pp. 1181–1198, 1996.

[7] D. Fish, A. Brinicombe, and E. Pike, "Blind deconvolution by means of the Richardson–Lucy algorithm," *J. Opt. Soc. Amer. A,* vol. 12, Jan. 1996.

[8] R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images.* Boston, MA: Kluwer, 1991.

[9] R. L. Lagendijk, A. M. Tekalp, and J. Biemond, "Maximum likelihood image and blur identification: A unifying approach," *Opt. Eng.,* vol. 29, p. 4, 1990.

[10] R. L. Lagendijk, J. Biemond, and D. E. Boekee, "Identification and restoration of noisy blurred images using the expectation-maximization algorithm," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 1180–1191, July 1990.

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

[12] C. Vogel and M. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci. Stat. Comput.,* vol. 17, pp. 227–238, 1996.

[13] Y. You and M. Kaveh, "A regularization approach to joint blur identification and image restoration," *IEEE Trans. Image Processing,* vol. 5, pp. 416–428, Mar. 1996.

[14] ——, "Anisotropic blind image restoration," in *IEEE Int. Conf. Image Processing,* Lausanne, Switzerland, 1996.

Tony F. Chan, for a photograph and biography, see this issue, p. 309.

Chiu-Kwong Wong received the M.Phil. degree in mathematics from the University of Hong Kong in 1994. He has since been a doctoral student at the University of California, Los Angeles. His research interests include image restoration, signal detection, numerical linear algebra, and numerical PDE's.