An Iteratively Reweighted Norm Algorithm for Total Variation Regularization

Paul Rodríguez and Brendt Wohlberg

***Abstract*— Total Variation (TV) regularization has become a popular method for a wide variety of image restoration problems, including denoising and deconvolution. Recently, a number of authors have noted the advantages, including superior performance with certain non-Gaussian noise, of replacing the standard ℓ^2 data fidelity term with an ℓ^1 norm. We propose a simple but very flexible and computationally efficient method, the Iteratively Reweighted Norm algorithm, for minimizing a generalized TV functional which includes both the ℓ^2-TV and and ℓ^1-TV problems.**

I. Introduction

Total Variation (TV) regularization was first introduced for image denoising [1], and has since evolved into a more general tool for solving a wide variety of image restoration problems, including deconvolution and inpainting [2], [3]. The ℓ^2-TV regularized solution of the inverse problem involving data $\mathbf{b}$ and forward linear operator A (the identity in the case of denoising, and a convolution for a deconvolution problem, for example) is the minimum of the functional

$$J(\mathbf{u}) = \frac{1}{2}\left\| A\mathbf{u} - \mathbf{b} \right\|_2^2 + \lambda \left\| \sqrt{(D_x\mathbf{u})^2 + (D_y\mathbf{u})^2} \right\|_1 \qquad (1)$$

where $\left\| \sqrt{(D_x\mathbf{u})^2 + (D_y\mathbf{u})^2} \right\|_1$ is the total variation of $\mathbf{u}$, and D_x, D_y denote the horizontal and vertical discrete derivative operators respectively. While a variety of algorithms [4], [5] have been proposed to solve this optimization problem, it remains a computationally expensive task which can be prohibitively costly for large problems and non-sparse forward operator A.

Recently, the modified TV functional with an ℓ^1 data fidelity term

$$J(\mathbf{u}) = \left\| A\mathbf{u} - \mathbf{b} \right\|_1 + \lambda \left\| \sqrt{(D_x\mathbf{u})^2 + (D_y\mathbf{u})^2} \right\|_1 \qquad (2)$$

has attracted attention [6], [7] due to a number of advantages, including superior denoising performance with salt and pepper (speckle) noise [8]. The standard approaches to solving problem (1) are not effective for problem (2), for which efficient (time-performance and memory requirements) algorithm development is not well advanced [7].

In this paper, we introduce the Iteratively Reweighted Norm (IRN) algorithm for solving the generalized TV functional

$$J(\mathbf{u}) = \frac{1}{p}\left\| A\mathbf{u} - \mathbf{b} \right\|_p^p + \frac{\lambda}{q} \left\| \sqrt{(D_x\mathbf{u})^2 + (D_y\mathbf{u})^2} \right\|_q^q \qquad (3)$$

for $p \geq 1$ and $q \geq 1$ (we have found the algorithm to converge for smaller p and q values, but cannot prove that the algorithm converges in these cases). The IRN algorithm is a simple but computationally efficient and very flexible method which is competitive with existing, well-established [4], [5] algorithms for ℓ^2-TV, and while we have not compared it with all of the ℓ^1-TV algorithms of which we are aware [7], [8], [9], [10], [11], it is significantly faster than those with which we have performed comparisons, and in contrast to many of these other methods, easily allows the inclusion of forward operator A for a more general inverse problem than denoising.

II. Iteratively Reweighted Norm Approach

A. Previous Related Work

The IRN approach is closely related to the Iteratively Reweighted Least Squares (IRLS) [12], [13] method, which minimizes the ℓ^p norm

$$J(\mathbf{u}) = \frac{1}{p}\left\| A\mathbf{u} - \mathbf{b} \right\|_p^p \qquad (4)$$

by approximating it, within an iterative scheme, by a weighted ℓ^2 norm problem, for which an algebraic solution is easily derived. (It is also worth mentioning similar methods used for constructing sparse signal representations [14], [15].) A brief outline of the convergence proof for IRLS provides a useful introduction to the outline of the IRN convergence proof presented in Section II-D. At each iteration k, define the functional

$$Q^{(k)}(\mathbf{u}) = \frac{1}{2}\left\| W^{(k)^{1/2}} \left(A\mathbf{u} - \mathbf{b} \right) \right\|_2^2 + \left(1 - \frac{p}{2}\right) J(\mathbf{u}^{(k)}),$$

where $W^{(k)} = \operatorname{diag}\left(|A\mathbf{u}^{(k)} - \mathbf{b}|^{p-2} \right)$, and $\mathbf{u}^{(k)}$ is the current solution (i.e., the solution obtained at the end of the previous iteration). It may be shown that

$$\begin{aligned} J(\mathbf{u}) &\leq Q^{(k)}(\mathbf{u}) \quad \forall \mathbf{u} \\ J(\mathbf{u}^{(k)}) &= Q^{(k)}(\mathbf{u}^{(k)}) \\ \nabla J(\mathbf{u}^{(k)}) &= \nabla Q^{(k)}(\mathbf{u}^{(k)}), \end{aligned}$$

i.e. the quadratic functional $Q^{(k)}(\mathbf{u})$ is tangent to $J(\mathbf{u})$ at $\mathbf{u} = \mathbf{u}^{(k)}$, where it is also an upper bound for $J(\mathbf{u})$. As a

Paul Rodríguez is with T-7 Mathematical Modeling and Analysis, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: prodrig@t7.lanl.gov, Tel: (505) 606 1483 , Fax: (505) 665 5757

Brendt Wohlberg is with T-7 Mathematical Modeling and Analysis, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: brendt@t7.lanl.gov, Tel: (505) 667 6886, Fax: (505) 665 5757

This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and was partially supported by the NNSA's Laboratory Directed Research and Development Program.

result [13], the iteration

$$\mathbf{u}^{(k+1)} = \left(A^T W^{(k)} A\right)^{-1} A^T \mathbf{b}, \tag{5}$$

which minimizes $Q^{(k)}(\mathbf{u})$ using the weights derived from the previous iteration, converges to the minimizer of $J(\mathbf{u})$.

When $p < 2$, the definition $W^{(k)} = \operatorname{diag}\left(|A\mathbf{u}^{(k)} - \mathbf{b}|^{p-2}\right)$ must be modified to avoid the possibility of division by zero. For $p = 1$, it may be shown (see [16]) that the choice

$$W^{(k)}_{n,n} = \begin{cases} |r^{(k)}_n|^{-1} & \text{if } |r^{(k)}_n| \geq \epsilon \\ \epsilon^{-1} & \text{if } |r^{(k)}_n| < \epsilon \end{cases},$$

where $\mathbf{r}^{(k)} = A\mathbf{u}^{(k)} - \mathbf{b}$, and ϵ is a small positive number, guarantees global convergence to the minimizer of $\sum_n \rho_\epsilon(r_n)$, where

$$\rho_\epsilon(r_n) = \begin{cases} \epsilon^{-1} r_n^2 & \text{if } |r_n| \leq \epsilon^2 \\ 2|r_n| - \epsilon & \text{if } |r_n| > \epsilon^2 \end{cases}$$

is the Huber function. We will address a similar question for the IRN algorithm in the following sections.

B. IRN: Data Fidelity Term

The data fidelity term of Equation (3) has the form of the IRLS functional in Equation (4), and is handled in the same way, representing

$$\frac{1}{p}\left\|A\mathbf{u} - \mathbf{b}\right\|_p^p \quad \text{by} \quad \frac{1}{2}\left\|W_F^{1/2}\left(A\mathbf{u} - \mathbf{b}\right)\right\|_2^2$$

with iteratively updated weights W_F. Since the choice of W_F (as described in Section II-A) gives infinite weights for zero-valued components of $A\mathbf{u} - \mathbf{b}$, when $p < 2$, we set

$$W_F = \operatorname{diag}\left(\frac{2}{p} f_F(A\mathbf{u} - \mathbf{b})\right),$$

where f_F is defined (for some small ϵ_F) as

$$f_F(x) = \begin{cases} |x|^{p-2} & \text{if } |x| > \epsilon_F \\ \epsilon_F^{p-2} & \text{if } |x| \leq \epsilon_F, \end{cases}$$

which is a common approach for IRLS algorithms [17].

C. IRN: Regularization Term

It is not quite as obvious how to express the TV regularization term from Equation (3) as a weighted ℓ^2 norm. Given vectors $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ we have (using block-matrix notation)

$$\left\|\begin{pmatrix} W_R^{1/2} & 0 \\ 0 & W_R^{1/2} \end{pmatrix}\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\chi} \end{pmatrix}\right\|_2^2 = \sum_k w_k \xi_k^2 + w_k \chi_k^2$$

so that when

$$W_R = \operatorname{diag}\left(\frac{2}{q}\left(\boldsymbol{\xi}^2 + \boldsymbol{\chi}^2\right)^{(q-2)/2}\right)$$

we have

$$\frac{1}{2}\left\|\begin{pmatrix} W_R^{1/2} & 0 \\ 0 & W_R^{1/2} \end{pmatrix}\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\chi} \end{pmatrix}\right\|_2^2 = \frac{1}{q}\left\|\sqrt{\boldsymbol{\xi}^2 + \boldsymbol{\chi}^2}\right\|_q^q.$$

We therefore set $\boldsymbol{\xi} = D_x\mathbf{u}$, $\boldsymbol{\chi} = D_y\mathbf{u}$, and define the operator D and weights $\tilde{W}_R$

$$D = \begin{pmatrix} D_x \\ D_y \end{pmatrix} \quad \tilde{W}_R = \begin{pmatrix} W_R & 0 \\ 0 & W_R \end{pmatrix}$$

so that $\|\tilde{W_R}^{1/2} D\mathbf{u}\|_2^2 = \|W_R^{1/2} D_x\mathbf{u}\|_2^2 + \|W_R^{1/2} D_y\mathbf{u}\|_2^2$ with weights defined by

$$W_R = \operatorname{diag}\left(\frac{2}{q}\left((D_x\mathbf{u})^2 + (D_y\mathbf{u})^2\right)^{(q-2)/2}\right)$$

gives the desired term. (Note that this is *not* the separable approximation, as in [18], for example, which is often used.)

As in the case of the data fidelity term, care needs to be taken when $q < 2$ and $D_x\mathbf{u}^2 + D_y\mathbf{u}^2$ has zero-valued components. We define

$$f_R(x) = \begin{cases} |x|^{(q-2)/2} & \text{if } |x| > \epsilon_R \\ 0 & \text{if } |x| \leq \epsilon_R \end{cases}$$

for some small ϵ_R, and set

$$W_R = \operatorname{diag}\left(\frac{2}{q} f_R\left((D_x\mathbf{u})^2 + (D_y\mathbf{u})^2\right)\right).$$

Note that f_R sets values smaller than the threshold, ϵ_R, to zero, as opposed to f_F, which sets values smaller than the threshold, ϵ_F, to ϵ_F^{p-2}. The motivation for this choice is that a region with very small or zero gradient should be allowed to have zero contribution to the regularization term, rather than be clamped to some minimum value. In practice, however, we have found that this choice does not give significantly different results than the standard IRLS approach represented by f_F.

D. Convergence of the IRN algorithm

Here we briefly sketch the proof of global convergence of the IRN algorithm. Combining the terms described in Sections II-B and II-C, we define the generalized TV functional

$$\begin{aligned} Q^{(k)}(\mathbf{u}) \;=\; & \frac{1}{2}\left\|W_F^{(k)^{1/2}}\left(A\mathbf{u} - \mathbf{b}\right)\right\|_2^2 + \frac{\lambda}{2}\left\|\tilde{W}_R^{(k)^{1/2}} D\mathbf{u}\right\|_2^2 + \\ & \left(1 - \frac{p}{2}\right) F(\mathbf{u}^{(k)}) + \lambda\left(1 - \frac{q}{2}\right) R(\mathbf{u}^{(k)}), \end{aligned} \tag{6}$$

where $W_F^{(k)}$, $W_R^{(k)}$, and $\mathbf{u}^{(k)}$ are the current data fidelity weights, regularization weights, and solution respectively. The constant (with respect to $\mathbf{u}$) terms

$$\begin{aligned} F(\mathbf{u}^{(k)}) &= \left\|A\mathbf{u}^{(k)} - \mathbf{b}\right\|_p^p \\ R(\mathbf{u}^{(k)}) &= \left\|\sqrt{(D_x\mathbf{u}^{(k)})^2 + (D_y\mathbf{u}^{(k)})^2}\right\|_q^q \end{aligned}$$

do not effect the IRN iteration since they have zero gradient with respect to $\mathbf{u}$, but are necessary so that $J(\mathbf{u}^{(k)}) = Q^{(k)}(\mathbf{u}^{(k)})$. (Note that $J(\mathbf{u}^{(k)}) = \frac{1}{p}F(\mathbf{u}^{(k)}) + \frac{\lambda}{q}R(\mathbf{u}^{(k)})$, as in Equation (3).)

It is easily shown that

$$\begin{aligned} J(\mathbf{u}) &\leq Q^{(k)}(\mathbf{u}) \quad \forall \mathbf{u} \\ J(\mathbf{u}^{(k)}) &= Q^{(k)}(\mathbf{u}^{(k)}) \\ \nabla J(\mathbf{u}^{(k)}) &= \nabla Q^{(k)}(\mathbf{u}^{(k)}) \\ \nabla^2 Q^{(k)}(\mathbf{u}) &= A^T W_F^{(k)} A + \lambda D^T \tilde{W}_R^{(k)} D \;>\; 0 \end{aligned}$$

Fig. 1. 512×512 Lena image.

As in the IRLS case (Section II-A), the quadratic functional $Q^{(k)}(\mathbf{u})$ is tangent to $J(\mathbf{u})$ at $\mathbf{u} = \mathbf{u}^{(k)}$, where it is also an upper bound for $J(\mathbf{u})$, and it has a positive definite Hessian. Using these results, it can be shown that the minimizer of $Q^{(k)}(\mathbf{u})$, given by

$$\mathbf{u} = \left(A^T W_F^{(k)} A + \lambda D^T \tilde{W}_R^{(k)} D\right)^{-1} A^T W_F^{(k)} \mathbf{b}, \qquad (7)$$

converges to the minimizer of (3) as we iterate over k.

III. IRN Algorithms

The IRN algorithm can be easily derived (see Algorithm 1) from Equation (7). The matrix inversion can be achieved using the Conjugate Gradient (CG) or Preconditioned CG (PCG) method such as Jacobi line relaxation (JLR) or symmetric Gauss-Seidel line relaxation (SLGS) [19]. We have also found that a significant speed improvement may be achieved by starting with a high CG/PCG tolerance, which is decreased with each main iteration until the final desired value is reached. For the results reported hereafter, operators D_x and D_y are defined by applying the same one-dimensional discrete derivative along image rows and columns respectively. Applied to vector $\mathbf{u} \in \mathbb{R}^N$, this discrete one-dimensional derivative is computed as $u_k - u_{k+1}$ for the derivative at index $k \in \{0, 1, \ldots, N-2\}$, and as $(u_{N-3} - 8u_{N-2})/12$ for the derivative at index $N-1$. Values in the ranges 10^{-2} to 10^{-4} and 10^{-4} to 10^{-8} were used for constants ϵ_F and ϵ_R respectively. All program run times were obtained using the NUMIPAD library [20] on a (Linux) 3.0GHz Intel Pentium-4 processor.

In the remainder of this paper we shall restrict our attention to the l^1-TV case ($p = 1$, $q = 1$), but note that this flexible approach is capable of efficiently solving other cases as well, including the standard l^2-TV case ($p = 2$, $q = 1$) where we have found it to have similar performance to the lagged diffusivity algorithm [4]; for a 512×512 input image (Figure 1), normalized between 0 and 1, corrupted with Gaussian noise ($\sigma = 0.05$) lagged diffusivity required 1.69s for 5 iterations, giving a denoised image with 16.24 dB, while the ℓ^2-TV IRN (via PCG-SLGS) required 1.69s for 5 iterations, giving a 16.02 dB output. The number of CG iterations (per main loop) needed by lagged-diffusivity and several flavors of the IRN algorithm are shown in Figure 2.

Algorithm 1 IRN algorithm - General case

Initialize

$$\mathbf{u}^{(0)} = \left(A^T A + \lambda D^T D\right)^{-1} A^T \mathbf{b}$$

Iterate

$$W_F^{(k)} = \operatorname{diag}\left(\frac{2}{p} f_F(A\mathbf{u}^{(k-1)} - \mathbf{b})\right)$$

$$W_R^{(k)} = \operatorname{diag}\left(\frac{2}{q} f_R\left((D_x \mathbf{u}^{(k-1)})^2 + (D_y \mathbf{u}^{(k-1)})^2\right)\right)$$

$$\mathbf{u}^{(k)} = \Big(A^T W_F^{(k)} A + \lambda D_x^T W_R^{(k)} D_x + \lambda D_y^T W_R^{(k)} D_y\Big)^{-1} A^T W_F^{(k)} \mathbf{b}$$

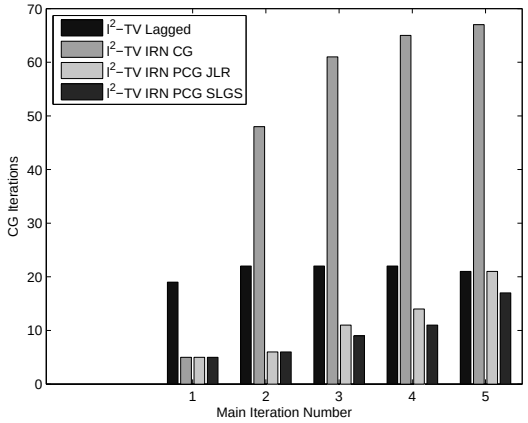

Fig. 2. A comparison of CG iterations for lagged-diffusivity and several flavors of the IRN algorithm for ℓ^2-TV denoising. The IRN algorithm is slow when coupled with a standard CG solver, but the use of preconditioned CG (PCG) strategies, such as Jacobi line relaxation (JLR) and symmetric Gauss-Seidel line relaxation (SLGS), dramatically reduces the number of required CG iterations.

A. IRN ℓ^1-TV Denoising

Direct application of Equation (7) for ℓ^1-TV denoising is slow due to the ill-conditioning of the system to be inverted. However, in this case we may apply the substitution $\tilde{\mathbf{u}} = W_F^{(k)^{1/2}} \mathbf{u}$ to Equation (7), giving

$$\tilde{\mathbf{u}} = \left(I + \lambda W_F^{(k)^{-1/2}} D^T \tilde{W}_R^{(k)^{1/2}} D W_F^{(k)^{-1/2}}\right)^{-1} W_F^{(k)^{1/2}} \mathbf{b}.$$

Applying this modification (*indirect* IRN algorithm) to the general (*direct*) algorithm results in a significant reduction in the required number of CG iterations; moreover if a PCG strategy is applied (e.g. JLR or SLGS), the number of CG iterations is further reduced, as shown in Figure 4, where a comparison between the number of CG iteration for 5 loops for the direct and indirect CG, PCG-JLR and PCG-SLGS IRN algorithms is carried out to denoise a 512×512 image (Figure 1) corrupted with 5% speckle noise. The run times were 8.2s, 2.2s, 1.3s

(a) Image with 20% speckle noise. SNR: -1.76dB.

(b) Denoised ℓ^1-TV ($\lambda = 1.25$). SNR: 17.42dB. Time: 1.63s

(c) Image with 50% speckle noise. SNR: -5.77dB.

(d) Denoised ℓ^1-TV ($\lambda = 1.25$). SNR: 9.60dB. Time: 2.60s

Fig. 3. Extreme examples for ℓ^1-TV denoising using the IRN algorithm: original image (Figure 1) was corrupted with 20% and 50% speckle noise.

and 1.3s respectively, with denoised qualities of 17.36 dB, 21.85 dB, 20.35 dB, and 20.36 dB respectively. More extreme examples are presented in Figure 3, with the original image corrupted by 20% (Figure 3(a)) and 50% (Figure 3(c)) speckle noise. The result of ℓ^1-TV denoising these images, using the IRN (via PCG-JLR) algorithm, is surprisingly good: 17.4 dB (Figure 3(b)) in 1.63s when denoising Figure 3(a) and 9.6 dB (Figure 3(d)) in 2.6s when denoising Figure 3(c).

B. IRN ℓ^1-TV Deconvolution

We apply the IRN algorithm to the problem of deconvolution of an image convolved by a separable smoothing filter having 9 taps, approximating a Gaussian with standard deviation of 2.0. In this case A is the corresponding linear operator, and the substitution applied in the previous section is no longer possible. We constructed a test image by convolving the image in Figure 1 by the smoothing kernel, and adding 5% speckle noise (see Figure 5(a)), giving an image with an SNR of 4.2dB. Comparing the performance of ℓ^2-TV (via the lagged diffusivity algorithm) and ℓ^1-TV (via the IRN method) deconvolution, we obtain a 12.2dB reconstruction in 9.2s and a 14.4dB reconstruction in 26.8s respectively (see Figure 5(b)).

IV. Conclusions

The Iteratively Reweighted Norm (IRN) approach provides a simple but computationally efficient method for TV regularized optimization problems, including both denoising and those such as deconvolution having a linear operator in the data fidelity term. This framework is very flexible, and can be applied to regularized inversions with a wide variety of norms for the data fidelity and regularization terms, including the standard l^2-TV, and more recently proposed l^1-TV formulations. This method provides a significantly faster algorithm for the l^1-TV formulation than any other algorithm of which we are aware. The NUMIPAD library [20], is proving to be a useful tool for solving optimization problems and can be used to reproduce the results presented in this paper.

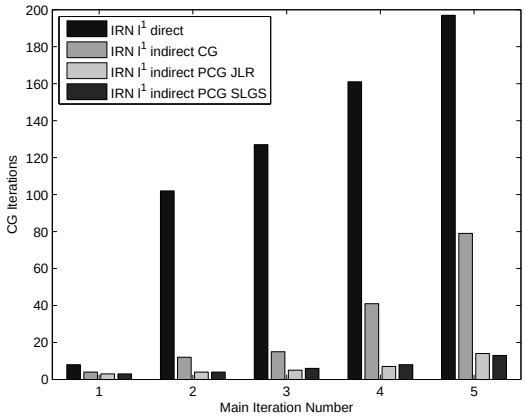

Fig. 4. A comparison of CG iterations for the direct and several flavors of the indirect IRN algorithm for ℓ^1-TV denoising. Both preconditioned CG (PCG) strategies, Jacobi line relaxation (JLR) and symmetric Gauss-Seidel line relaxation (SLGS), dramatically reduced the number of required CG iterations.

ACKNOWLEDGMENT

The authors thank Markus Berndt for valuable discussion on CG preconditioning methods.

REFERENCES

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

[2] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Transactions on Image Processing*, vol. 7, pp. 813–824, June 1998.

[3] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," in *The Handbook of Mathematical Models in Computer Vision* (N. Paragios, Y. Chen, and O. Faugeras., eds.), Springer, 2005.

[4] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM Journal on Scientific Computing*, vol. 17, pp. 227–238, Jan. 1996.

[5] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004.

[6] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers," *SIAM Journal on Numerical Analysis*, vol. 40, no. 3, pp. 965–994, 2002.

[7] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized l^1 function approximation," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817–1837, 2005.

[8] M. Nikolova, "A variational approach to remove outliers and impulse noise," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 99–120, 2004. 4th Conference on Mathematics and Image Analysis; September 10-13, 2002; Paris, France.

[9] J. Darbon and M. Sigelle, "A fast and exact algorithm for total variation minimization," *Lecture Notes in Computer Science*, vol. 3522/2005, pp. 351–359, 2005.

[10] D. Goldfarb and W. Yin, "Second-order cone programming methods for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 27, no. 2, pp. 622–645, 2005.

[11] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition - modeling, algorithms, and parameter selection," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 111–136, 2006.

[12] R. Wolke and H. Schwetlick, "Iteratively reweighted least squares: Algorithms, convergence analysis, and numerical comparisons," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, pp. 907–921, Sept. 1988.

[13] K. P. Bube and R. T. Langan, "Hybrid ℓ^1/ℓ^2 minimization with applications to tomography," *Geophysics*, vol. 62, pp. 1183–1195, July-August 1997.

[14] I. F. Gorodnitsky and B. D. Rao, "A new iterative weighted norm minimization algorithm and its applications," in *IEEE Sixth SP Workshop on Statistical Signal and Array Processing, 1992*, (Victoria, BC, Canada), Oct. 1992.

(a) Blurred image with 5% speckle noise. SNR: 4.2dB.

(b) Deblurred ℓ^1-TV ($\lambda = 1.3$). SNR: 14.4dB. Time: 26.8s

Fig. 5. ℓ^1-TV deconvolution using the IRN algorithm. The original image (Figure 1) was convolved by a separable smoothing filter before the addition of 5% speckle noise.

[15] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Transactions On Signal Processing*, vol. 47, pp. 187–200, Jan. 1999.

[16] S. A. Ruzinsky and E. T. Olsen, "L_1 and L_∞ minimization via a variant of Karmarkar's algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 245–253, 1989.

[17] J. A. Scales and A. Gersztenkorn, "Robust methods in inverse theory," *Inverse Problems*, vol. 4, pp. 1071–1091, Oct. 1988.

[18] Y. Li and F. Santosa, "A computational algorithm for minimizing total variation in image restoration," *IEEE Transactions on Image Processing*, vol. 5, pp. 987–995, June 1996.

[19] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*. SIAM Books, 2nd ed., 2000.

[20] P. Rodríguez and B. Wohlberg, "Numerical methods for inverse problems and adaptive decomposition (NUMIPAD)." Software library available from `http://numipad.sourceforge.net/`.