

# PERFORMANCE COMPARISON OF ITERATIVE REWEIGHTING METHODS FOR TOTAL VARIATION REGULARIZATION

*Paul Rodríguez\**

*Brendt Wohlberg†*

Electrical Department  
Pontificia Universidad Católica del Perú  
Lima, Peru

Theoretical Division  
Los Alamos National Laboratory  
Los Alamos, NM 87545, USA

## ABSTRACT

Iteratively Reweighted Least Squares (IRLS) is a well-established method of optimizing  $\ell^p$  norm problems such as Total Variation (TV) regularization. Within this general framework, there are several possible ways of constructing the weights and the form of the linear system that is iteratively solved as part of the algorithm. Many of these choices are equally reasonable from a theoretical perspective, and there has, thus far, been no systematic comparison between them. In this paper we provide such a comparison between the main choices in IRLS algorithms for  $\ell^1$ - and  $\ell^2$ -TV denoising, finding that there is a significant variation in the computational cost and reconstruction quality of the different variants.

**Index Terms**— Total Variation, Iteratively Reweighted Least Squares, Iteratively Reweighted Norm

## 1. INTRODUCTION

The generalized Total Variation (TV) denoising functional is

$$T(\mathbf{u}) = \frac{1}{p} \|\mathbf{u} - \mathbf{b}\|_p^p + \frac{\lambda}{q} \|\nabla \mathbf{u}\|_q, \quad (1)$$

where  $p = 2, q = 1$  and  $p = 1, q = 1$  correspond to the well-known cases of  $\ell^2$ -TV [1] and  $\ell^1$ -TV [2] respectively. There are numerous algorithms for minimizing (1) that are based on the Euler-Lagrange equation (for a recent review, see [3]), and use a smooth approximation of

$$\|\nabla \mathbf{u}\|_1 \approx \frac{1}{2} \sum_n \tau_\epsilon(|(\nabla \mathbf{u})_n|), \quad (2)$$

where  $(\nabla \mathbf{u})_n$  represents the  $n^{\text{th}}$  term of  $\nabla \mathbf{u}$ ; for example, a popular choice is  $\tau_\epsilon(x) = 2\sqrt{x^2 + \epsilon^2}$  [1, 4], [2, Sec. 4]. Of these algorithms, we focus here on those that can be derived within the Iteratively Reweighted Least Squares (IRLS) framework, which are simple to implement, and have been

shown to be competitive [5] with more recent algorithms such as those based on the Split-Bregman (SB) [6] or Alternating Direction Method of Multipliers (ADMM) [7, 8] methods.

These algorithms approximate (1) via

$$Q^{(k)}(\mathbf{u}) = \frac{1}{2} \|W_F^{(k)0.5}(\mathbf{u} - \mathbf{b})\|_2^2 + \frac{\lambda}{2} \|W_R^{(k)0.5} D\mathbf{u}\|_2^2 \quad (3)$$

where  $D = [D_x^T D_y^T]^T$ , with  $D_x$  and  $D_y$  denoting the horizontal and vertical discrete derivative operators, and  $W_F^{(k)}$  and  $W_R^{(k)}$  are the weighting diagonal matrices used to iteratively approximate (1) by a weighted quadratic functional. The linear system obtained by setting the gradient

$$\nabla Q(\mathbf{u}) = W_F \mathbf{u} - W_F \mathbf{b} + \lambda D^T W_R D \mathbf{u}, \quad (4)$$

to zero is usually solved via CG (conjugate gradient) or PCG (preconditioned CG).

Within this general framework, there are numerous possible choices, equally reasonable from a theoretical perspective, for (i) the definition of the weighting matrices  $W_F$  and  $W_R$  to avoid division by zero problems, and (ii) the form of (4) that is actually solved. Although a wide variety of these choices have been considered, there has not been any systematic comparison of their relative performance. The goal of the present paper is to provide such a comparison so that future implementers of algorithms within this widely-used framework are able to make informed decisions in making these choices.

## 2. $\ell^1$ - AND $\ell^2$ -TV IRLS TAXONOMY

IRLS-based algorithms for  $\ell^1$ - and  $\ell^2$ -TV seek to solve (3), for which it is straightforward to compute the gradient (4). Following the standard IRLS scheme, the primary computation consist of solving the linear system resulting from setting  $\nabla Q(\mathbf{u}) = 0$ . However, there are a number of possibilities for the specific form of the linear system to be solved; the three main options (independent of how the weights are chosen) will be explored in Section 2.2. We also explore the options for choosing the weights in (3) in Section 2.1. Finally, while it is out of the scope of this work, we also mention that it

\*This research was supported by PUCP's "Fondo Concursable DARI-DGI" Program.

†This research was supported by the NNSA's Laboratory Directed Research and Development Program.

is possible to spatially adapt the regularization parameter for  $\ell^2$ -TV [9, 10, 11] and  $\ell^1$ -TV [12, 13, 14].

## 2.1. Weight Construction

Straightforward computation of the fidelity weights leads to

$$W_F^{(k)} = \text{diag} \left( |\mathbf{u}^{(k)} - \mathbf{b}|^{p-2} \right). \quad (5)$$

Clearly, when  $p < 2$ , there exists the possibility of a division by zero when computing  $W_F^{(k)}$ . For  $p = 1$  in particular, setting

$$W_F^{(k)}(n, n) = \alpha_\epsilon(r_n^{(k)}) \quad (6)$$

where  $\mathbf{r}^{(k)} = \mathbf{u}^{(k)} - \mathbf{b}$ ,  $\epsilon$  is a small positive number and

$$\alpha_\epsilon(x) = \begin{cases} |x|^{-1} & \text{if } |x| \geq \epsilon \\ \epsilon^{-1} & \text{if } |x| < \epsilon \end{cases}, \quad (7)$$

is usually preferred since it guarantees global convergence for the IRLS case [15], although other approaches have been used, such as

- $W_F^{(k)} = \text{diag} \left( (|\mathbf{r}^{(k)}| + \epsilon)^{-1} \right)$  (e.g. see [16]), and
- the Huber function  $W_F^{(k)}(n, n) = v_\epsilon(r_n^{(k)})$ , with

$$v_\epsilon(x) = \begin{cases} \epsilon^{-1}x^2 & \text{if } |x| \leq \epsilon \\ 2|x| - \epsilon & \text{if } |x| > \epsilon \end{cases} \quad (8)$$

(see [17, (77)], [15] amongst others).

In the case of the regularization weights, a straightforward computation leads to  $W_R^{(k)} = I_2 \otimes \Omega^{(k)}$  where  $\otimes$  is the Kronecker product, and

$$\Omega_R^{(k)} = \text{diag} \left( (\mathbf{z}^{(k)})^{-1} \right), \quad \mathbf{z}^{(k)} = \sqrt{(D_x \mathbf{u}^{(k)})^2 + (D_y \mathbf{u}^{(k)})^2} \quad (9)$$

for which the possibility of a division by zero clearly also exists. Such a situation can be avoided by using (7), i.e.  $\Omega_R^{(k)}(n, n) = \alpha_\epsilon(z_n^{(k)})$ , or the closely related  $\Omega_R^{(k)}(n, n) = \beta_\epsilon(z_n^{(k)})$  (used in [18]) with

$$\beta_\epsilon(x) = \begin{cases} |x|^{-1} & \text{if } |x| > \epsilon \\ 0 & \text{if } |x| \leq \epsilon \end{cases}. \quad (10)$$

Other options include  $\Omega_R^{(k)} = \text{diag} \left( ((\mathbf{z}^{(k)})^2 + \epsilon^2)^{-0.5} \right)$  ([1, 4]) and the Huber function (used in [19], although it is seldom used in IRLS-based algorithms since the gradient of (3) depends on the derivative of the Huber function).

Summarizing, there are options that use the  $\epsilon$  value

**WT** as a threshold ((7) and (10)),

**WA** as an additive constant:  $\Omega_R^{(k)} = \text{diag} \left( ((\mathbf{z}^{(k)})^2 + \epsilon^2)^{-0.5} \right)$   
or  $W_F^{(k)} = \text{diag} \left( (|\mathbf{r}^{(k)}| + \epsilon)^{-1} \right)$ , or

**WS** as a parameter for a function that is smooth at zero, such as the Huber function.

In the WT case, the threshold may be fixed, or may be automatically adapted to the input image (we will use the subscripts ‘‘F’’ or ‘‘A’’ respectively), as was first proposed, along with (10), in [18, Sec. IV.G]. While adaptive setting of the WA and WS parameters is in principle possible, we are not aware of any published methods for doing so, and do not consider this option here.

## 2.2. Linear System Form

The straightforward solution of (4) gives the linear system

$$(W_F + \lambda D^T W_R D) \mathbf{u} = W_F \mathbf{b} \quad (11)$$

where we have dropped the  $k$  superscript to ease the notation; this linear system can be easily solved using CG or PCG.

However, when  $W_F$  is not the identity (as when  $p < 2$  in (3)), the change of variable  $\mathbf{v} = W_F^{0.5} \mathbf{u}$  can be used to give

$$\text{solve: } (I + \lambda W_F^{-0.5} D^T W_R D W_F^{-0.5}) \mathbf{v} = W_F^{0.5} \mathbf{b} \quad (12)$$

$$\text{compute: } \mathbf{u} = W_F^{-0.5} \mathbf{v}. \quad (13)$$

In this case, since the fidelity weights are raised to a negative power ( $W_F^{-0.5}$ ), computation of ((12)-(13)) does not incur in numerical instabilities when constructing  $W_F$  as in (5). This approach was first described in [18].

Alternatively, by noticing that the well-known matrix inversion lemma (MIL) can be applied to (11), we get  $\mathbf{u} = (W_F^{-1} + W_F^{-1} D^T (\frac{W_R^{-1}}{\lambda} + D W_F^{-1} D^T)^{-1} D W_F^{-1}) W_F \mathbf{b}$ , which can be efficiently solved via

$$\text{solve: } \left( \frac{W_R^{-1}}{\lambda} + D W_F^{-1} D^T \right) \mathbf{z} = D \mathbf{b} \quad (14)$$

$$\text{compute: } \mathbf{u} = \mathbf{b} - W_F^{-1} D^T \mathbf{z}. \quad (15)$$

Once again, since the fidelity and regularization weights are raised to a negative power, their direct computation ((5) and (9) respectively) do not incur in numerical instabilities. This approach was first described in [20].

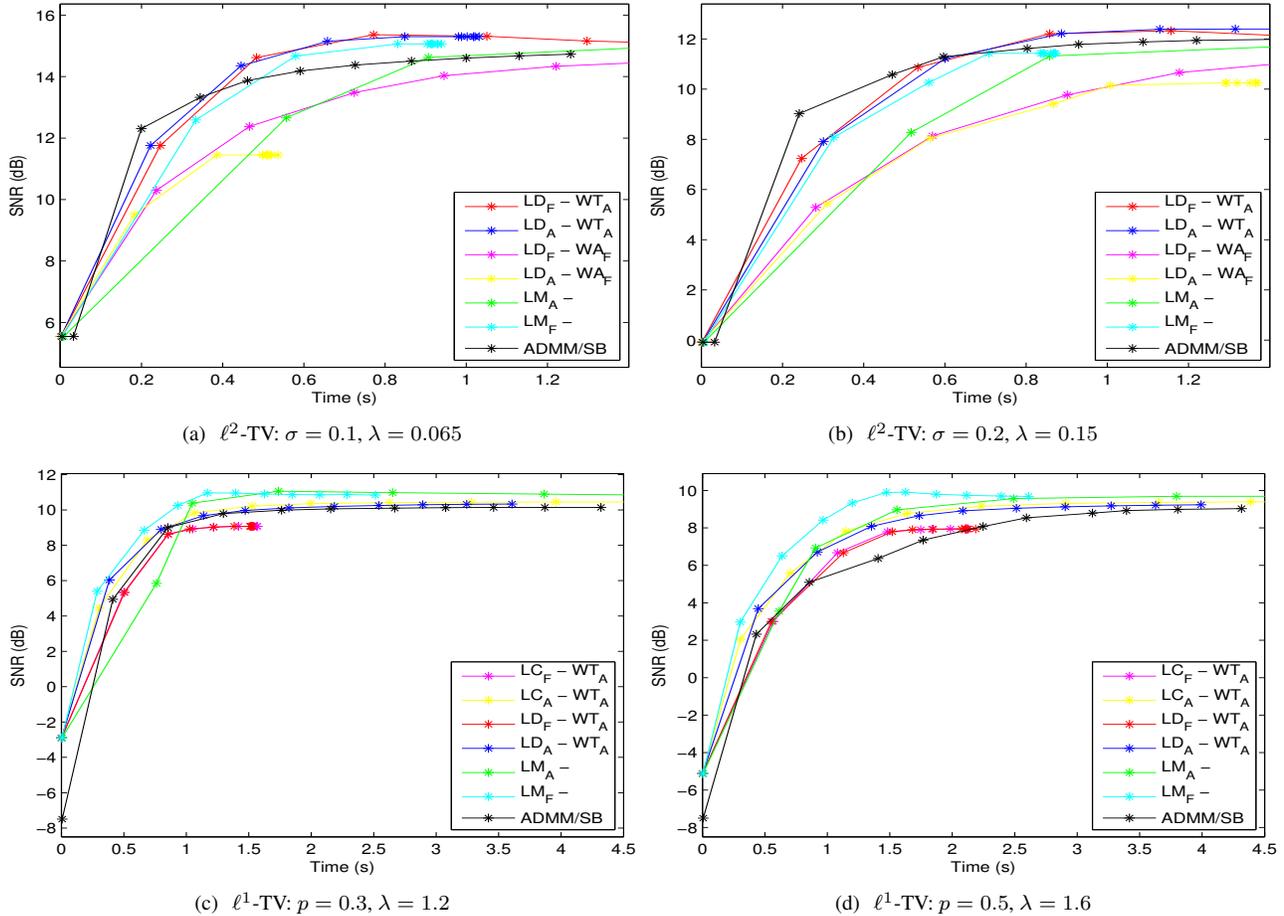
Summarizing, the linear system to be solved may be

**LD** directly as in (4),

**LC** constructed via change of variable, as in (12)-(13), or

**LM** derived via application of the MIL, as in (14)-(15).

Each of these systems can be solved to a fixed accuracy, or using an adaptive accuracy at each outer iteration (for which we will also use the subscripts ‘‘F’’ or ‘‘A’’ respectively).



**Fig. 1.** Evolution of  $\ell^2$ - and  $\ell^1$ -TV denoising quality against time for different schemes. The Lena test image was corrupted with Gaussian noise with (a)  $\sigma = 0.1$ , denoised with  $\lambda = 0.065$ , and (b)  $\sigma = 0.2$ , denoised with  $\lambda = 0.15$ , and the Barbara test image was corrupted with salt & pepper noise with (c)  $p = 0.3$ , denoised with  $\lambda = 1.2$ , and (d)  $p = 0.5$ , denoised with  $\lambda = 1.6$ .

### 3. EXPERIMENTAL RESULTS

In order to compare the computational performance of the different methods (listed in Section 2) and SB/ADMM algorithm for TV denoising, we have implemented all the described algorithms in Matlab code. All simulations have been carried out using the above mentioned code on a 1.73GHz Intel core i7 CPU laptop (L2: 6144K, RAM: 6G).

Due to space constraints we only present a subset of the results, but all other simulations can be found at [21]. The ground-truth test image, scaled between zero and one (Lena, Barbara, or Cameraman) is denoted by  $\mathbf{u}^*$ . The observed image  $\mathbf{b}$  is corrupted with additive Gaussian noise  $\mathbf{b} = \mathbf{u}^* + \sigma \cdot \boldsymbol{\eta}$  where  $\sigma \in \{0.05, 0.1, 0.2\}$  and  $\boldsymbol{\eta}$  is unit variance Gaussian noise, or with salt & pepper (SNP) noise  $\mathbf{b} = \mathcal{B} \cdot \mathbf{u}^* + (1 - \mathcal{B}) \cdot \mathbf{r}$  with  $\mathcal{B}$  being drawn from an i.i.d. multivariate Bernoulli distribution with success probability  $1 - p$  for  $p = \{0.1, 0.3, 0.5\}$ , and  $\mathbf{r}$  being SNP noise.

We present results for the WT and WA weight variants (see Section 2.1), and for the LD, LC (where appropriate) and

LM linear system variants (see Section 2.2). Comparisons are also provided with the  $\ell^2$ -TV or  $\ell^1$ -TV SB/ADMM algorithm with a fixed ADMM penalty parameter(s) set to  $2\lambda$  and  $10\lambda$  respectively (for other options see [22, Section 3.4.1]) as a general baseline.

All WT results are of the  $WT_A$  variety since we have previously found this to provide superior performance to  $WT_F$  [18, Sec. IV.G]). The histogram thresholds for these adaptive settings were 1% for regularization terms and 5% for the data fidelity term of  $\ell^1$ -TV. For the  $WA_F$  variant (we do not have a readily available candidate algorithm for adapting it) we used  $\epsilon = 10^{-3}$  since  $\epsilon = 10^{-4}$  was much slower without a significant increase in reconstruction quality.

When using the  $LD_F$  (11) and  $LM_F$  (12)-(13) variants with fixed CG/PCG accuracy we have set the tolerance to  $10^{-4}$ , since it gives a good compromise between speed and reconstruction quality. Similarly, we have observed good performance for the  $LC_F$  variant (14)-(15) even for a rather loose accuracy parameter (for the experiments reported here we have set it to  $10^{-1}$ ).

The following procedure is used for the cases where the accuracy is adapted at each main iteration. At iteration  $k$ , the adaptive accuracy is based on a relative residual computed from the previous solution  $\mathbf{u}^{(k-1)}$ , the previous tolerance  $\tau^{(k-1)}$  and the current weights  $W_F^{(k)}$  and  $W_R^{(k)}$ . For instance, in the case of the LD variant we have

$$\rho = \|(W_F^{(k)} + \lambda D^T W_R^{(k)} D)\mathbf{u}^{(k-1)} - W_F^{(k)}\mathbf{b}\|_2 / \|\mathbf{b}\|_2 \quad (16)$$

and the new tolerance is set to  $\tau^{(k)} = \frac{\tau^{(k-1)}}{c}$ , where  $c > 1$  is a constant (we use  $c = 5$ ). A similar residual can be constructed for the LM variant.

For each variant we compute 10 outer iterations, although in Fig. 1 we limit the time axis to 1.2 and 4.5 seconds for  $\ell^2$ -TV or  $\ell^1$ -TV respectively to capture the general behavior and avoid emphasizing variants with poor time performance.

In Figs. 1(a) and 1(b) we present the SNR with respect to ground truth against time for the  $(512 \times 512)$  grayscale Lena image corrupted with Gaussian noise with  $\sigma = \{0.1, 0.2\}$  for some of the variants described in Section 2, as well as for the  $\ell^2$ -TV SB/ADMM. Except for the  $LM_A$  and  $WA_F$  variants, all other cases (including the  $\ell^2$ -TV SB/ADMM) have a very similar performance in terms of reconstruction quality and computational performance; this in accordance with the main result of [5] (IRLS base algorithm are competitive with SB/ADMM algorithm).

In Figs. 1(c) and 1(d) we present the SNR with respect to ground truth against time for the  $512 \times 512$  grayscale Barbara image corrupted with SNP noise with  $p = \{0.3, 0.5\}$  for all the  $\ell^1$ -TV variants described in Section 2, as well as for  $\ell^1$ -TV SB/ADMM. It is interesting to note that the relatively obscure variant  $LM_F$  clearly outperforms all others in terms of computational speed and reconstruction quality. Although we do not present results for  $p = 0.1$  due to space constraints, our experiments shows that for that level of noise there is no real difference between all the considered implementations.

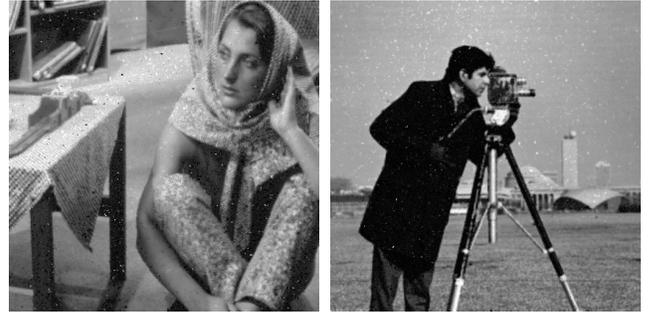
In Fig. 4 we compare the reconstruction quality of the  $LM_F$  and  $LC_A$  IRLS variants and the ADMM/SB algorithm. The original image (either Barbara or Cameraman) was corrupted with SNP noise with  $p = 0.5$  and was reconstructed using 10 main iterations of the IRLS variants as well as with the ADMM/SB algorithm, all of them using the same regularization parameter ( $\lambda = 1.6$ ). While some isolated artifacts can be observed for the denoised images via the  $LM_F$  variant (Figs. 4(a) and 4(b)), overall they are better than the denoised images via the  $LC_A$  variant (Figs. 4(c)-(d)) or SB/ADMM algorithm (Figs. 4(e)-(f)).

#### 4. CONCLUSIONS

In this paper we have presented a systematic comparison of the different possible choices for constructing the weights and forming the linear systems among different  $\ell^p$ -TV IRLS-based variants, with the SB/ADMM algorithm used as a general baseline.

In the  $\ell^2$ -TV case, the  $WT_A$  weight construction variant appears to enjoy a substantial advantage over  $WA_F$ . The  $LD_F$  and  $LD_A$  methods provide very similar performance here, but the  $LD_A$  strategy is preferable given the dependence of  $LD_F$  on manual parameter tuning. Overall, the  $LD_A$ - $WT_A$  variant provides the best performance for this case.

In the case of  $\ell^1$ -TV the seldom used  $LM_F$  variant, resulting from applying the matrix inversion lemma, which in the case of IRLS-based TV avoids numerical instabilities in the computation of the associated weights, has a superior performance in terms of both computational cost and reconstruction quality for medium to high levels of salt & pepper noise.



(a) SNR: 9.76dB, SSIM: 0.67 (b) SNR: 15.95dB, SSIM: 0.83



(c) SNR: 9.28dB, SSIM: 0.62 (d) SNR: 13.13dB, SSIM: 0.83



(e) SNR: 8.94dB, SSIM: 0.57 (f) SNR: 12.21dB, SSIM: 0.74

**Fig. 2.**  $\ell^1$ -TV denoising results for the test images Barbara and Cameraman corrupted with  $\sigma = 0.5$  and denoised with  $\lambda = 1.6$ . Results were obtained by IRN variant  $LM_F$  in (a)-(b), IRN variant  $LC_A$  in (c)-(d), and SB/ADMM in (e)-(f).

## 5. 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] M. Nikolova, "A variational approach to remove outliers and impulse noise," *J. of Math. Imaging and Vision*, vol. 20, pp. 99–120, 2004.
- [3] P. Rodríguez, "Total variation regularization algorithms for images corrupted with different noise models: A review," *Journal of Electrical and Computer Engineering*, vol. 2013, no. Article ID 217021, 2013.
- [4] C. Vogel and M. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Trans. Image Proc.*, vol. 7, no. 6, pp. 813–824, Jun. 1998.
- [5] P. Rodríguez and B. Wohlberg, "A comparison of the computational performance of iteratively reweighted least squares and alternating minimization algorithms for  $\ell_1$  inverse problems," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 3069–3072.
- [6] T. Goldstein and S. Osher, "The Split Bregman method for L1-regularized problems," *SIAM J. Img. Sci.*, vol. 2, pp. 323–343, Apr. 2009.
- [7] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Applic.*, vol. 2, no. 1, pp. 1416–1438, 1976.
- [8] R. Glowinski and A. Marrocco, "Sur l'approximation par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *Revue Fr. Autom., Inform., Rech. Opér. Analyse numérique*, vol. 9, no. 2, pp. 41–72, 1975.
- [9] D. Strong and T. Chan, "Spatially and scale adaptive total variation based regularization and anisotropic diffusion in image processing," UCLA Math Department, CAM Report CAM 96-46, Nov. 1996.
- [10] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. on Image Processing*, vol. 17, no. 9, pp. 1540–1554, 2008.
- [11] Y. Lin, B. Wohlberg, and H. Guo, "UPRE method for total variation parameter selection," *Elsevier Signal Processing*, vol. 90, no. 8, pp. 2546–2551, Aug. 2010.
- [12] R. Chan, C. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Trans. Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.
- [13] Y. Dong, M. Hintermüller, and M. Rincon-Camacho, "Automated regularization parameter selection in multi-scale total variation models for image restoration," *J. of Math. Imag. and Vision*, pp. 1–23, Dec. 2010.
- [14] R. Rojas and P. Rodríguez, "Spatially adaptive total variation image denoising under salt and pepper noise," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, 2011, pp. 314–318.
- [15] S. Ruzinsky and E. Olsen, " $L_1$  and  $L_\infty$  minimization via a variant of Karmarkar's algorithm," *IEEE Trans. ASSP*, vol. 37, pp. 245–253, Feb. 1989.
- [16] L. Bar, N. Kiryati, and N. Sochen, "Image deblurring in the presence of impulsive noise," *International Journal of Computer Vision*, vol. 70, no. 3, pp. 279–298, December 2006.
- [17] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers," *SIAM J. Numerical Analysis*, vol. 40, no. 3, pp. 965–994, 2002.
- [18] P. Rodríguez and B. Wohlberg, "Efficient minimization method for a generalized total variation functional," *IEEE Trans. Image Proc.*, vol. 18, no. 2, pp. 322–332, Feb. 2009.
- [19] C. Vogel, *Computational Methods for Inverse Problems*. SIAM, 2002.
- [20] M. Figueiredo, J. Bioucas-Dias, J. Oliveira, and R. Nowa, "On total-variation denoising: A new majorization-minimization algorithm and an experimental comparison with wavelet denoising," in *Proceedings of the International Conference on Image Processing (ICIP)*, Atlanta, GA, USA, Oct. 2006, pp. 2633–2636.
- [21] P. Rodríguez and B. Wohlberg, "Numerical methods for inverse problems and adaptive decomposition (NUMIPAD)," <http://numipad.sourceforge.net/>.
- [22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Fdns. & Trends in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.