LA-UR- 07-7881

Approved for public release; distribution is unlimited.

Title: CONTRAST INVARIANCE OF THE ITERATIVELY REWEIGHTED NORM ALGORITHM FOR L1-TV REGULARIZATION

Author(s): BRENDT WOHLBERG
PAUL RODRIGUEZ

Intended for: Proceedings:
2008 IEEE SOUTHWEST SYMPOSIUM ON IMAGE ANALYSIS AND INTERPETATION

Los Alamos
NATIONAL LABORATORY
EST.1943

Form 836 (7/06)

Contrast Invariance of the Iteratively Reweighted Norm Algorithm for ℓ^1-TV Regularization

Paul Rodríguez and Brendt Wohlberg

Abstract— **It is well-known that ℓ^1 Total Variation (TV) denoising is contrast invariant in the sense that the minimizer for an input multiplied by some scalar is the the same scalar multiplied by the minimizer for the unscaled input. While the recently introduced Iteratively Reweighted Norm algorithm for minimizing TV functionals preserves this property in principle, we have observed that, in practice, as a result of finite-precision arithmetic, significantly difference results may be obtained by scaling the input. We discuss some relevant implementational details, and describe a modified algorithm which avoids problems arising from dependence on the contrast of the input.**

Index Terms— **image restoration, inverse problem, regularization, total variation**

I. Introduction

The standard TV regularized solution of the denoising problem involving data $\mathbf{b}$ is the minimum of the functional

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

for $p=2$, $q=1$ (referred as ℓ^2-TV) and where we employ the following notation:

- $F(\mathbf{u}) = \frac{1}{p}\left\|\mathbf{u}-\mathbf{b}\right\|_p^p$ is the data fidelity term.
- $R(\mathbf{u}) = \frac{1}{q}\left\|\sqrt{(D_x\mathbf{u})^2+(D_y\mathbf{u})^2}\right\|_q^q$ is the regularization term,
- the p-norm of vector $\mathbf{u}$ is denoted by $\|\mathbf{u}\|_p$,
- scalar operations applied to a vector are considered to be applied element-wise, so that, for example, $\mathbf{u}=\mathbf{v}^2 \Rightarrow u_k = v_k^2$ and $\mathbf{u}=\sqrt{\mathbf{v}} \Rightarrow u_k = \sqrt{v_k}$, and
- horizontal and vertical discrete derivative operators are denoted by D_x and D_y respectively.

An important modification has been to use the ℓ^1 norm as the fidelity term, corresponding to choosing $p=1$, $q=1$ in (1). This modified functional (referred as ℓ^1-TV), was first analyzed in [1], [2], [3] (see also [4] for a good tutorial).

Recently, the Iteratively Reweighted Norm algorithm [5], [6], [7] was proposed for minimizing the generalized TV functional of Equation (1), which includes the ℓ^2-TV and ℓ^1-TV as special cases, by representing the ℓ^p and ℓ^q norms by the equivalent weighted ℓ^2 norms. In particular, the IRN algorithm provides a very fast algorithm for the ℓ^1-TV case, where it is competitive with the state of the art algorithms for ℓ^1-TV denoising ([8], [9], [10], [11]).

The contrast invariant property (see [3], [4]) in ℓ^1-TV denoising is understood as follows:

Property 1.1: if $\widehat{\mathbf{u}}^*$ is the minimizer to Eq. 1 for observed input $\widehat{\mathbf{b}}$ then $\mathbf{u}^* = \beta\widehat{\mathbf{u}}^*$ is the minimizer to Eq. 1 for observed input $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$, where $\beta > 0$.

The IRN algorithm does comply with this property, nevertheless we have observed numerical artifacts for $\beta \gg 1$ (i.e. $\beta = 255$ and $\widehat{\mathbf{b}} \in [0..1]$) and the computational results do not follow the theory. In section III we propose a simple modification to the IRN algorithm to correct its computational behavior.

This paper is organized as follows: in Section II we briefly described the IRN algorithm with focus on computational issues. Next, in Section III, we describe the numerical artifacts of the IRN algorithm for ℓ^1-TV denoising and how to correct them. Finally we show numerical results Section IV and list our concluding remarks in Section V.

II. Iteratively Reweighted Norm Approach

The IRN algorithm approximates the Total Variation functional (Eq. (1)) by a ℓ^2 weighted version of the original one, and when solved iteratively, it converges to the solution given by Eq. (1). The new functional is given by:

$$T^{(k)}(\mathbf{u}) = \frac{1}{2}\left\|W_F^{(k)^{1/2}}(\mathbf{u}-\mathbf{b})\right\|_2^2 + \frac{\lambda}{2}\left\|W_R^{(k)^{1/2}}D\mathbf{u}\right\|_2^2 \qquad (2)$$

where

$$W_F^{(k)} = \operatorname{diag}\left(\tau_F(\mathbf{u}^{(k)}-\mathbf{b})\right), \qquad (3)$$

and τ_F is defined (for some small ϵ_F) as

$$\tau_F(x) = \begin{cases} |x|^{p-2} & \text{if } |x| > \epsilon_F \\ \epsilon_F^{p-2} & \text{if } |x| \le \epsilon_F, \end{cases} \qquad (4)$$

to avoid numerical problems for $p<2$ and $\mathbf{u}^{(k)}-\mathbf{b}=0$; this strategy is a common approach for IRLS type algorithms [12]. The matrices D and $W_R^{(k)}$ (regularization term) are defined by

$$D = \begin{pmatrix} D_x \\ D_y \end{pmatrix} \quad W_R^{(k)} = \begin{pmatrix} \Omega_R^{(k)} & 0 \\ 0 & \Omega_R^{(k)} \end{pmatrix} \qquad (5)$$

where

$$\tau_R(x) = \begin{cases} |x|^{(q-2)/2} & \text{if } |x| > \epsilon_R \\ 0 & \text{if } |x| \le \epsilon_R \end{cases} \qquad (6)$$

Paul Rodríguez is with Digital Signal Processing Group at the Pontific Catholic University of Peru, Lima, Peru. Email: prodrig@pucp.edu.pe, Tel: (+51 1) 9339-5427

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: (+1 505) 667 6886, Fax: (+1 505) 665 5757

for some small ϵ_R, and set

$$\Omega_R^{(k)} = \mathrm{diag}\left(\tau_R\left((D_x\mathbf{u}^{(k)})^2 + (D_y\mathbf{u}^{(k)})^2\right)\right), \quad (7)$$

An iteratively minimization of Eq. 2 results in an algorithm that finds $\mathbf{u}^{(k)}$

$$\mathbf{u}^{(k)} = \left(W_F^{(k)} + \lambda D^T W_R^{(k)} D\right)^{-1} W_F^{(k)}\mathbf{b} \quad (8)$$

for $k = 1, 2, ..,$ with $\mathbf{u}^{(0)} = \left(I + \lambda D^T D\right)^{-1}\mathbf{b}$ as the initial solution.

The matrix inversion can be achieved using the Conjugate Gradient (CG) or the Preconditioned CG (PCG) method such as Jacobi line relaxation (JLR) or symmetric Gauss-Seidel line relaxation (SLGS) [13].

For two dimensional datasets the matrix $W_F^{(k)} + \lambda D^T W_R^{(k)} D$ is block-diagonal and has a very specific structure: it has a main block (tridiagonal) and two off-diagonals. It is convenient to store it as two tridiagonal matrices: without loss of generality let the input data be stored in row-major then the two tridiagonal matrices will be given by $W_F^{(k)} + \lambda D_x^T \Omega_R^{(k)} D_x$ and $\lambda D_y^T \Omega_R^{(k)} D_y$ and note that the first matrix will operate over contiguous elements in memory whereas the second one will operate over elements with a stride equal to the number of columns of the input dataset. This storage strategy allows a very efficient implementation of the matrix times vector operation, heavily used in CG or PCG.

For $p < 2$ (see Eq. (1)) we may apply the substitution $\mathbf{v} = W_F^{1/2}\mathbf{u}$ in Equation (2) giving

$$T^{(k)}(\mathbf{v}) = \frac{1}{2}\left\|\mathbf{v} - W_F^{(k)^{1/2}}\mathbf{b}\right\|_2^2 + \frac{\lambda}{2}\left\|W_R^{(k)^{1/2}} D W_F^{(k)^{-1/2}}\mathbf{v}\right\|_2^2,$$

with solution

$$\mathbf{v}^{(k)} = \left(I + \lambda W_F^{(k)^{-1/2}} D^T W_R^{(k)} D W_F^{(k)^{-1/2}}\right)^{-1} W_F^{(k)^{1/2}}\mathbf{b}. \quad (9)$$

In this case the resulting algorithm will compute $\mathbf{v}^{(k)}$ and then $\mathbf{u}^{(k)} = W_F^{(k)^{-1/2}}\mathbf{v}^{(k)}$; similarly $\mathbf{u}^{(0)} = \left(I + \lambda D^T D\right)^{-1}\mathbf{b}$ is used as the initial solution.

The substitution $\mathbf{v} = W_F^{1/2}\mathbf{u}$, specially in the ℓ^1-TV denoising case, was found to result in a very large reduction in the required number of CG iterations and in increased reconstruction quality (SNR).

Note that the condition numbers of the matrices to be inverted in equations (8) and (9) are (theoretically) the same, but computationally they behave different. Heuristically the product $W_F^{(k)^{-1/2}} D^T W_R^{(k)} D W_F^{(k)^{-1/2}}$ reduces the dynamic range of the matrix to be inverted and therefore reduces the cumulative error due to finite precision computations (matrix times vector in the CG or PCG solver).

The strategy to compute the product $I + \lambda W_F^{(k)^{-1/2}} D^T W_R^{(k)} D W_F^{(k)^{-1/2}}$ times a given vector is to pre-compute two tridiagonal matrices $I + \lambda W_F^{(k)^{-1/2}} D_x^T \Omega_R^{(k)} D_x W_F^{(k)^{-1/2}}$, which will operate over contiguous elements and $\lambda W_F^{(k)^{-1/2}} D_y^T \Omega_R^{(k)} D_y W_F^{(k)^{-1/2}}$ which will operate over elements with a stride equal to the number of columns of the input dataset.

Note that for both cases (equations (8) and (9)) a line relaxation preconditioning strategy can be easily implemented due to the storage policy. For instance, in equation (9) the relaxation will be given by $R = I + \lambda W_F^{(k)^{-1/2}} D_x^T \Omega_R^{(k)} D_x W_F^{(k)^{-1/2}} + \mathrm{diag}(\lambda W_F^{(k)^{-1/2}} D_y^T \Omega_R^{(k)} D_y)$.

III. Contrast Invariant ℓ^1-TV IRN Algorithm

In this section we analyze the compliance to the contrast invariant property (property 1.1) by the IRN algorithm for ℓ^1-TV denoising. We focus in the case when $\beta \gg 1$ (i.e. $\beta = 255$) and $\widehat{\mathbf{b}} \in [0..1]$, typical if the input image has been acquired using a digital still camera or a CCD video camera.

Let $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$ be the noisy input, where β is a positive constant and $\widehat{\mathbf{b}} \in [0..1]$. Without loss of generality we focus in the case when $\beta \gg 1$. Let $\widehat{\epsilon}_F$ and $\widehat{\epsilon}_R$ be the thresholds used for $\widehat{\mathbf{b}} \in [0..1]$ and $\epsilon_F = \beta\widehat{\epsilon}_F$ and $\epsilon_R = \beta\widehat{\epsilon}_R$ be the thresholds used for $\mathbf{b} \in [0..\beta]$ then it is straightforward to show that for the ℓ^1-TV case

$$\begin{aligned} \tau_F(\mathbf{u}^{(k)} - \mathbf{b}) &= \frac{1}{\beta}\tau_F(\widehat{\mathbf{u}}^{(k)} - \widehat{\mathbf{b}}) \\ \tau_R\left((D_x\mathbf{u}^{(k)})^2 + (D_y\mathbf{u}^{(k)})^2\right) &= \tfrac{1}{\beta}\tau_R\left((D_x\widehat{\mathbf{u}}^{(k)})^2 + (D_y\widehat{\mathbf{u}}^{(k)})^2\right), \end{aligned}$$

where $\mathbf{u}^{(k)}$ and $\widehat{\mathbf{u}}^{(k)}$ are the k^{th} solution (Eq. (8) or (9)) for inputs $\mathbf{b}$ and $\widehat{\mathbf{b}}$ respectively.

The weighting matrices used to denoise $\widehat{\mathbf{b}}$ will be defined by (using $\widehat{\epsilon}_F$ and $\widehat{\epsilon}_R$ as the thresholds)

$$\widehat{W}_F^{(k)} = \mathrm{diag}\left(\tau_F(\widehat{\mathbf{u}}^{(k)} - \widehat{\mathbf{b}})\right) \quad (10)$$

$$\widehat{\Omega}_R^{(k)} = \mathrm{diag}\left(\tau_R\left((D_x\widehat{\mathbf{u}}^{(k)})^2 + (D_y\widehat{\mathbf{u}}^{(k)})^2\right)\right) \quad (11)$$

and $\widehat{W}_R^{(k)}$ follows from Eq. (5). Furthermore, from equations (3), (7) and (10), (11) we note a similar equivalence for the weighting matrices used to denoise $\mathbf{b}$ and $\widehat{\mathbf{b}}$

$$W_F^{(k)} = \frac{1}{\beta}\widehat{W}_F^{(k)} \qquad W_R^{(k)} = \frac{1}{\beta}\widehat{W}_R^{(k)}. \quad (12)$$

Eq. (12) implies that for equivalent threshold levels and $\beta \gg 1$, the numerical values of the weighting matrices $W_F^{(k)}$ and $W_R^{(k)}$ (numerical values of the input dataset $\mathbf{b}$ are in the range $[0..\beta]$) are much smaller than the numerical values of the weighting matrices $\widehat{W}_F^{(k)}$ and $\widehat{W}_R^{(k)}$ (numerical values of the input dataset $\widehat{\mathbf{b}}$ are in the range $[0..1]$). We will show that these differences in the weighting matrices (Eq. (12)) do not modify the linear system to be inverted (equations (8) and (9)) by the IRN algorithm, and it (theoretically) complies with the contrast invariant property. Nevertheless due to finite precision computations and straightforward arithmetic used by the IRN algorithm (as presented in [6]) when computing the weighting matrices, the computational results do not exhibit the the contrast invariant property; this can be easily corrected by reordering the arithmetic (to avoid numerical errors) when computing the weighting matrices

To show that IRN algorithm complies with the contrast invariant property, we first replace $W_F^{(k)}$ and $W_R^{(k)}$ by $\widehat{W}_F^{(k)}$

and $\widehat{W}_R^{(k)}$ in Eq. (8), then

$$\begin{aligned}\mathbf{u}^{(k)} &= \left(\frac{1}{\beta}\widehat{W}_F^{(k)} + \lambda D^T \frac{1}{\beta}\widehat{W}_R^{(k)} D\right)^{-1} \frac{1}{\beta}\widehat{W}_F^{(k)}\mathbf{b} \\ &= \left(\widehat{W}_F^{(k)} + \lambda D^T \widehat{W}_R^{(k)} D\right)^{-1} \widehat{W}_F^{(k)}\mathbf{b} \qquad (13)\end{aligned}$$

replacing $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$ and noting that $\widehat{\mathbf{u}}^{(k)} = \left(\widehat{W}_F^{(k)} + \lambda D^T \widehat{W}_R^{(k)} D\right)^{-1} \widehat{W}_F^{(k)}\widehat{\mathbf{b}}$ will be the solution to the ℓ^1-TV denoising problem with input $\widehat{\mathbf{b}}$ (instead of $\mathbf{b}$), is easy to conclude that $\mathbf{u}^{(k)} = \beta\,\widehat{\mathbf{u}}^{(k)}$. Similarly, in Eq. (9)

$$\begin{aligned}\mathbf{v}^{(k)} &= \left(I + \lambda\beta^{0.5}\widehat{W}_F^{(k)^{-0.5}} D^T \frac{1}{\beta}\widehat{W}_R^{(k)} D \beta^{0.5}\widehat{W}_F^{(k)^{-0.5}}\right)^{-1} \\ &\quad \frac{1}{\beta^{0.5}}\widehat{W}_F^{(k)^{0.5}}\mathbf{b} \\ &= \left(I + \lambda\widehat{W}_F^{(k)^{-0.5}} D^T \widehat{W}_R^{(k)} D \widehat{W}_F^{(k)^{-0.5}}\right)^{-1} \\ &\quad \frac{1}{\beta^{0.5}}\widehat{W}_F^{(k)^{0.5}}\mathbf{b} \qquad (14)\end{aligned}$$

then by setting $\widehat{\mathbf{v}}^{(k)} = \left(I + \lambda\widehat{W}_F^{(k)^{-0.5}} D^T \widehat{W}_R^{(k)} D \widehat{W}_F^{(k)^{-0.5}}\right)^{-1} \widehat{W}_F^{(k)^{0.5}}\widehat{\mathbf{b}}$, replacing $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$ (note that $\mathbf{v}^{(k)} = \beta^{0.5}\widehat{\mathbf{v}}^{(k)}$) and operating $\mathbf{u}^{(k)} = W_F^{(k)^{-0.5}}\mathbf{v}^{(k)}$ we get $\mathbf{u}^{(k)} = \beta^{0.5}\widehat{W}_F^{(k)^{-0.5}}\beta^{0.5}\widehat{\mathbf{v}}^{(k)} = \beta\,\widehat{\mathbf{u}}^{(k)}$

Comparing equations (8) and (9) with (13) and (14) we conclude that the linear system to be inverted (in any case) is the same whereas the data values of the input dataset are in the range $[0..\beta]$ or $[0..1]$; we also conclude that $\mathbf{u}^{(k)} = \beta\,\widehat{\mathbf{u}}^{(k)}$ and thus the contrast invariant property holds.

The IRN algorithm for ℓ^1-TV denoising, as presented in [6], directly solve for the linear system in Eq. (8) (or Eq. (9)) and thus it computes $W_F^{(k)}$ and $W_R^{(k)}$ without taking into account the scaling constant β and therefore the simplification (factoring out the constant value β) presented in equations (13) and (14) are explicit carried out (while inverting the linear system) using finite precision arithmetic, incurring in numerical errors that are observable: $\mathbf{u}^{(k)} \neq \beta\,\widehat{\mathbf{u}}^{(k)}$ for computational results achieved by ℓ^1-TV denoising (via the IRN algorithm) for noisy input data $\widehat{\mathbf{b}} \in [0..1]$ and $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$.

Note that equations (13) and (14) solve the ℓ^1-TV denoising problem for input $\mathbf{b}$ (not for $\widehat{\mathbf{b}}$), even though the weighting matrices (in the linear system needed to be inverted) are $\widehat{W}_F^{(k)}$ and $\widehat{W}_R^{(k)}$ (instead of $W_F^{(k)}$ and $W_R^{(k)}$).

This observation is the key to modify the IRN algorithm (as presented in [6]) in order to avoid the numerical errors (described above) and attain computational results that do exhibit the the contrast invariant property: the constant value β should be taken into account before solving the linear system. This is summarized in algorithm 1 which differs from the IRN algorithm as presented in [6] in that the former operates the matrix inversion using matrices $\widehat{W}_F^{(k)}$ and $\widehat{W}_R^{(k)}$ instead of matrices $W_F^{(k)}$ and $W_R^{(k)}$ (as done by the latter). In the case of shot noise, factor β can be directly computed as the maximum value of the noisy input $\mathbf{b}$; for other type of noise, an ad-hoc value should be estimated.

Inputs

$\mathbf{b}$: Noisy input dataset, where $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$ and $\widehat{\mathbf{b}} \in [0..1]$

Initialize

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

for $k = 1, 2, ...$

$$\widehat{W}_F^{(k)} = \mathrm{diag}\left(\beta \cdot \tau_F(\mathbf{u}^{(k-1)} - \mathbf{b})\right)$$

$$\widehat{\Omega}_R^{(k)} = \mathrm{diag}\left(\beta \cdot \tau_R\left((D_x\mathbf{u}^{(k-1)})^2 + (D_y\mathbf{u}^{(k-1)})^2\right)\right)$$

(i) $\mathbf{u}^{(k)} = \left(\widehat{W}_F^{(k)} + \lambda D^T \widehat{W}_R^{(k)} D\right)^{-1} \widehat{W}_F^{(k)}\mathbf{b}$

(ii-1) $\mathbf{v}^{(k)} = \left(I + \lambda\widehat{W}_F^{(k)^{-1/2}} D^T \widehat{W}_R^{(k)} D \widehat{W}_F^{(k)^{-1/2}}\right)^{-1} \widehat{W}_F^{(k)^{1/2}}\mathbf{b}$

(ii-2) $\mathbf{u}^{(k)} = \widehat{W}_F^{(k)^{-1/2}}\mathbf{v}^{(k)}$

end

Algorithm 1: Contrast Invariant IRN Algorithm. Matrices D and $W_R^{(k)}$ are defined in Equation (5). Step (i) refers to equation (8) and (13). Steps (ii) refers to equations (9) and (14).

IV. Numerical Results

Given a noisy input $\mathbf{b} = \beta\,\widehat{\mathbf{b}}$, where $\beta \gg 1$ and $\widehat{\mathbf{b}} \in [0..1]$ there are up to three different approaches to denoise it via IRN ℓ^1-TV:

(A1) use $\mathbf{b}$ to solve for Eq. (8) or (9) and find $\mathbf{u}$.

(A2) use $\widehat{\mathbf{b}}$ (instead of $\mathbf{b}$) to solve for Eq. (8) or (9), find $\widehat{\mathbf{u}}$ and then compute $\mathbf{u} = \beta\,\widehat{\mathbf{u}}$.

(A3) use $\mathbf{b}$ to solve for Eq. (13) or (14) and find $\mathbf{u}$. This is the case of the contrast invariant IRN algorithm.

In most of the cases, IRN-(A1) with $\beta = 255$ Eq. (8) or (9) do not converge to a stable solution, and therefore we will concentrate our simulations for cases IRN-(A2) and IRN-(A3). The test image will the Lena image (512×512 pixel); similar results are attained for other classical images (Barbara or Mandrill, often used in image processing) but not presented here due to space constrains. Except where specified otherwise, program run times were obtained on a 1.8GHz Intel dual-core processor with 2048K L2 cache and 1G of RAM. A software implementation [14], which can reproduce the results presented in this paper, is available under an open-source license.

In Figure 1 we compare the reconstruction qualities (SNR with respect to the reference noise-free image) for IRN-(A2) ℓ^1-TV and the Contrast Invariant IRN-(A3) ℓ^1-TV, where the input (noisy) image is Lena, corrupted with speckle noise (10% of its pixels, SNR 1.2dB). It is observed that for equations (9) and (14) (labeled in Figure 1 as IRN-(A2) via (9) and IRN-(A3) via (14) respectively, see also Figure 2) the Contrast Invariant property holds up to the fourth iteration when they start showing a small difference in the reconstruction quality (due to finite precision errors), having IRN-(A3) via (14) the best quality; this was expected since, by design, it reduces the computational errors.

When equations (8) and (13) are used, we observed that (i) when compared to equations (9) and (14) they present a

smaller reconstruction quality (Figure 1) and the number of CG iterations (Figure 3) needed to solve the linear system (at each iteration) is greater than the ones needed for equations (9) and (14) (this results are compatible with previously reported ones in [5], [6], [7]) and (ii) computationally they do not exhibit the contrast invariant property.

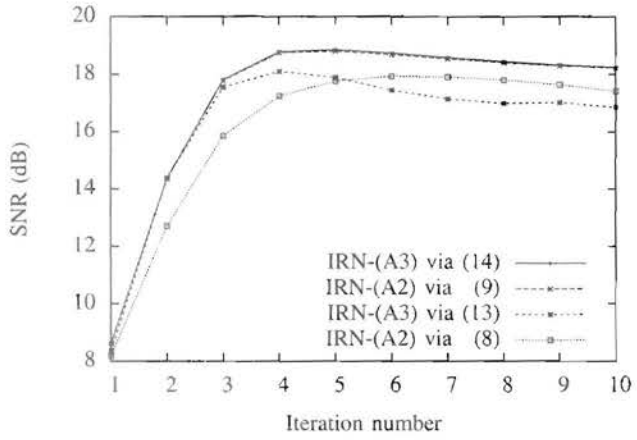

Fig. 1. ℓ^1-TV denoising SNR values against algorithm iteration number for $\lambda = 1.25$ using a fixed (10^{-4}) CG tolerance for all cases. Input image was Lena corrupted with speckle noise (10% of its pixels, SNR 1.2dB.).

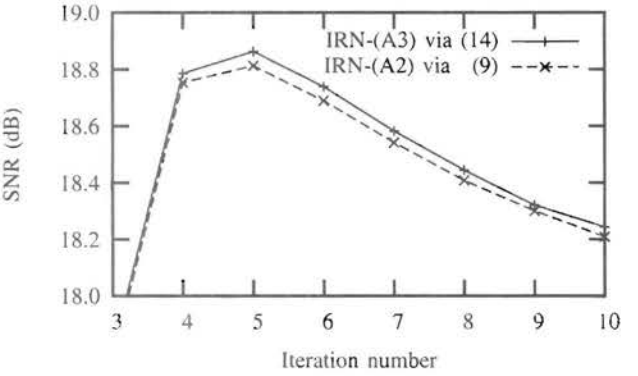

Fig. 2. A detail comparison for IRN-(A3) via (14) and IRN-(A2) via (9), corresponding to the simulation described for Figure 1.

V. Conclusions

We have presented a modified version of the IRN algorithm which preserves invariance to contrast of the input.

References

[1] S. Alliney, "Digital filters as absolute norm regularizers," *IEEE Transactions on Signal Processing*, vol. 40, no. 6, p. 15481562, 1992.

[2] ——, "A property of the minimum vectors of a regularizing functional defined by means of the absolute norm," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 913–917, 1997.

[3] 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.

[4] 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.

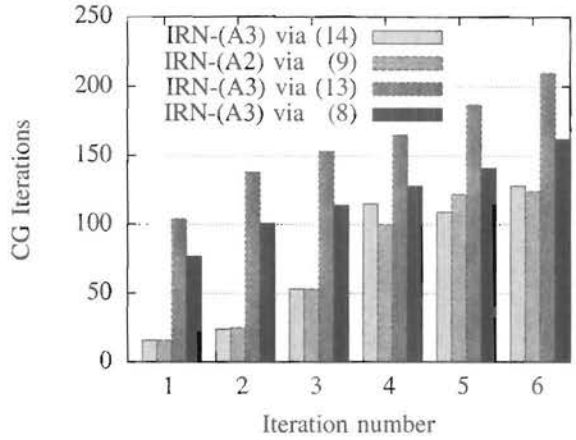

Fig. 3. A comparison of CG iterations, corresponding to the simulation described for Figure 1.

[5] P. Rodríguez and B. Wohlberg, "An iteratively weighted norm algorithm for total variation regularization," in *Proceedings of the 2006 Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 2006, pp. 892–896.

[6] B. Wohlberg and P. Rodríguez, "An iteratively reweighted norm algorithm for minimization of total variation functionals," *IEEE Signal Processing Letters*, vol. 14, no. 12, pp. 948–951, Dec. 2007.

[7] P. Rodríguez and B. Wohlberg, "Efficient minimization method for the generalized total variation functional," submitted to *IEEE Transaction on Image Processing*.

[8] 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.

[9] J. Darbon and M. Sigelle, "Image restoration with discrete constrained total variation part I: Fast and exact optimization," *Journal of Mathematical Imaging and Vision*, vol. 26, no. 3, pp. 261–276, 2006.

[10] ——, "Image restoration with discrete constrained total variation part II: Levelable functions, convex priors and non-convex cases," *Journal of Mathematical Imaging and Vision*, vol. 26, no. 3, pp. 277–291, 2006.

[11] D. Goldfarb and W. Yin, "Parametric maximum flow algorithms for fast total variation minimization," Rice University, Tech. Rep. TR07-09, 2007.

[12] J. A. Scales and A. Gersztenkorn, "Robust methods in inverse theory," *Inverse Problems*, vol. 4, no. 4, pp. 1071–1091, Oct. 1988.

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

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