

A NON-NEGATIVE QUADRATIC PROGRAMMING APPROACH TO MINIMIZE THE GENERALIZED VECTOR-VALUED TOTAL VARIATION FUNCTIONAL

Paul Rodríguez

Digital Signal Processing Group, Pontificia Universidad Católica del Perú
Lima, Peru
phone: + (511) 626-2000 anx 4681, email: prodrig@pucp.edu.pe
web: http://sites.google.com/a/istec.net/prodrig

ABSTRACT

We propose a simple but flexible method for solving the generalized vector-valued TV (VTV) functional with a non-negativity constraint. One of the main features of this recursive algorithm is that it is based on multiplicative updates only and can be used to solve the denoising and deconvolution problems for vector-valued (color) images.

This algorithm is the vectorial extension of the IRN-NQP (Iteratively Reweighted Norm - Non-negative Quadratic Programming) algorithm [1] originally developed for scalar (grayscale) images, and to the best of our knowledge, it is the only algorithm that explicitly includes a non-negativity constraint for color images within the TV framework.

1. INTRODUCTION

The development of numerical algorithms for vector-valued regularization has recently attracted considerable interest [2, 3, 4, 5, 6, 7, 8, 9]. In particular the Total Variation (TV) minimization scheme for deblurring color images, first introduced in [10], is of special interest since it can handle the Gaussian noise model and the salt-and-pepper noise model.

The ℓ^p vector-valued TV (VTV) regularized solution (with coupled-channel regularization [11]) of the inverse problem involving color image data \mathbf{b} and forward linear operator A is the minimum of the functional

$$T(\mathbf{u}) = \frac{1}{p} \left\| A\mathbf{u} - \mathbf{b} \right\|_p^p + \frac{\lambda}{q} \left\| \sqrt{\sum_{n \in C} (D_x \mathbf{u}_n)^2 + (D_y \mathbf{u}_n)^2} \right\|_q^q, \quad (1)$$

where $n \in C = \{r, g, b\}$ (note that C could represent an arbitrary number of channels) and the deblurring of images corrupted with Gaussian (ℓ^2 -VTV case) and salt-and-pepper noise (ℓ^1 -VTV case) can be performed when $p = 2$, $q = 1$ and $p = 1$, $q = 1$ in (1) respectively. We use the following notation:

- \mathbf{u}_n ($n \in C$) is a 1-dimensional (column) or 1D vector that represents a 2D grayscale image obtained via any ordering (although the most reasonable choices are row-major or column-major) of the image pixel.
- $\mathbf{u} = [(\mathbf{u}_r)^T (\mathbf{u}_g)^T (\mathbf{u}_b)^T]^T$ is a 1D (column) vector that represents a 2D color image.
- $\frac{1}{p} \|A\mathbf{u} - \mathbf{b}\|_p^p$ is the data fidelity term. For the scope of this paper, the linear operator A is assumed to be decoupled, i.e.: A is a diagonal block matrix with elements A_n and $n \in C = \{r, g, b\}$; if A is coupled (interchannel blur) due to channel crosstalk, it is possible to reduced it to a diagonal block matrix via a similarity transformation [12, 13],

- $\frac{1}{q} \left\| \sqrt{\sum_{n \in C} (D_x \mathbf{u}_n)^2 + (D_y \mathbf{u}_n)^2} \right\|_q^q$ is the generalization of TV regularization to color images with coupled channels (see [11, Section 9], also used in [5, 7, 8]),
- 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 $\mathbf{u} = \sqrt{\mathbf{v}} \Rightarrow u[k] = \sqrt{v[k]}$,
- $\sqrt{\sum_{n \in C} (D_x \mathbf{u}_n)^2 + (D_y \mathbf{u}_n)^2}$ is the discretization of $|\nabla \mathbf{u}|$ for coupled channels (see [7, eq. (3)]), and
- horizontal and vertical discrete derivative operators are denoted by D_x and D_y respectively.

The main target in TV problems is the denoising/deblurring of digital images, either grayscale or color. The enforcement of a non-negativity constraint, i.e.: $\mathbf{u} \geq 0$, for the solution of (1) is not only physically meaningful in most of the cases: images acquired by digital cameras, MRI, CT, etc., it also improves the quality of the reconstruction (see [14]). Nevertheless, the non-negativity constraint is seldom considered in the practice, since it makes a hard problem even harder. For scalar (grayscale) images, only a handful of numerical algorithms ([15, Ch. 9] and more recently [1, 14, 16]) include a non-negativity constraint on the solution of the TV problem ($C = \{\text{gray}\}$ in (1)). Currently for vector-valued (color) images, to best of our knowledge, there is no algorithm that explicitly includes the non-negativity constraint within the TV framework.

Recently, the IRN-NQP algorithm [1] was proposed to solve the generalized TV problem with an additional non-negativity constraint for scalar (grayscale) images. In this paper we augment the scope of [1] by including the ability to handle vector-valued (color) images, resulting in the vector-valued IRN-NQP algorithm, which solves the problem

$$\min T(\mathbf{u}) \quad \text{s.t. } 0 \leq \mathbf{u} \leq \mathbf{v}_{\max}, \quad (2)$$

where $T(\mathbf{u})$ is defined as in (1), and A is assumed to be a decoupled linear operator, which includes the denoising case ($A = I$ in (1)).

The vector-valued IRN-NQP algorithm (Iteratively Reweighted Norm or IRN, Non-negative Quadratic Programming or NQP) starts by representing the ℓ^p and ℓ^q norms in (1) by the equivalent weighted ℓ^2 norms, in the same fashion as the vector-valued Iteratively Reweighted Norm (IRN) algorithm (see [9]), and then cast the resulting weighted ℓ^2 functional as a Non-negative Quadratic Programming problem (NQP, see [17]), which uses an update rule that only involves matrix vector multiplication. Finally, we stress that

our algorithm can handle any norm with $0 < p, q \leq 2$, including the ℓ^2 -VTV and ℓ^1 -VTV as special cases.

2. THE VECTOR-VALUED IRN-NQP ALGORITHM

In this section we summarized the derivation of the vector-valued IRN (Iteratively Reweighted Norm) [9] algorithm as well as the description of the NQP (Non-negative Quadratic Programming) [17] problem to finally describe the vector-valued IRN-NQP algorithm.

2.1 The Vector-valued Iteratively Reweighted Norm (IRN) Algorithm

The vector-valued IRN approach is an extension of the IRN algorithm [18], and is closely related to Iteratively Reweighted Least Squares (IRLS) method for scalar [19] and vector [12] valued problems.

The vector-valued IRN approach represents the ℓ^p and ℓ^q norms in (1) by the equivalent weighted ℓ^2 norms, resulting in (see [9] for details):

$$T^{(k)}(\mathbf{u}) = \frac{1}{2} \left\| W_F^{(k)1/2} (A\mathbf{u} - \mathbf{b}) \right\|_2^2 + \frac{\lambda}{2} \left\| W_R^{(k)1/2} D\mathbf{u} \right\|_2^2 + \zeta \quad (3)$$

where $\mathbf{u}^{(k)}$ is a constant representing the solution of the previous iteration, ζ is a constant value, I_N is a $N \times N$ identity matrix, \otimes is the Kronecker product, $C = \{r, g, b\}$ and

$$W_F^{(k)} = \text{diag} \left(\tau_{F, \varepsilon_F} (A\mathbf{u}^{(k)} - \mathbf{b}) \right), \quad (4)$$

$$D = I_3 \otimes [D_x^T D_y^T]^T \quad W_R^{(k)} = I_6 \otimes \Phi^{(k)}, \quad (5)$$

$$\Phi^{(k)} = \text{diag} \left(\tau_{R, \varepsilon_R} \left(\sum_{n \in C} (D_x \mathbf{u}_n^{(k)})^2 + (D_y \mathbf{u}_n^{(k)})^2 \right) \right). \quad (6)$$

Following a common strategy in IRLS type algorithms [20], the functions

$$\tau_{F, \varepsilon_F}(x) = \begin{cases} |x|^{p-2} & \text{if } |x| > \varepsilon_F \\ \varepsilon_F^{p-2} & \text{if } |x| \leq \varepsilon_F \end{cases}, \quad (7)$$

and

$$\tau_{R, \varepsilon_R}(x) = \begin{cases} |x|^{(q-2)/2} & \text{if } |x| > \varepsilon_R \\ \varepsilon_R^{(q-2)/2} & \text{if } |x| \leq \varepsilon_R \end{cases}, \quad (8)$$

are defined to avoid numerical problems when $p, q < 2$ and $A\mathbf{u}^{(k)} - \mathbf{b}$ or $\sum_{n \in C} (D_x \mathbf{u}_n^{(k)})^2 + (D_y \mathbf{u}_n^{(k)})^2$ has zero-valued components.

2.2 Vector-valued IRN as Iteratively Reweighted Least Squares

We observe that by defining $W^{(k)} = \begin{pmatrix} W_F^{(k)} & 0 \\ 0 & W_R^{(k)} \end{pmatrix}$, $\tilde{A} = \begin{pmatrix} A \\ \sqrt{\lambda} D \end{pmatrix}$, and $\tilde{\mathbf{b}} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}$, we can cast (3) as a standard IRLS problem:

$$T^{(k)}(\mathbf{u}) = \frac{1}{2} \left\| W^{1/2(k)} (\tilde{A}\mathbf{u} - \tilde{\mathbf{b}}) \right\|_2^2. \quad (9)$$

Note that we are neglecting the constant term, since it has no impact on the solution of the optimization problem at hands. Moreover, after algebraic operations, the minimization problem in (9) can be expressed as

$$\min_{\mathbf{u}} T^{(k)}(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \tilde{A}^T W^{(k)} \tilde{A} \mathbf{u} - (\tilde{A}^T W^{(k)} \tilde{\mathbf{b}})^T \mathbf{u}. \quad (10)$$

It is straightforward to check that the matrix $\tilde{A}^T W^{(k)} \tilde{A}$ is symmetric and positive definite, and therefore solving

$$(\tilde{A}^T W^{(k)} \tilde{A}) \mathbf{u}^{(k+1)} = (\tilde{A}^T W^{(k)} \tilde{\mathbf{b}}), \quad (11)$$

gives the minimum of (10), and converges (see [18] for details) to the minimum of (1) as the iterations proceeds.

2.3 Non-negative Quadratic Programming (NQP)

Recently [17] an interesting and quite simple algorithm has been proposed to solve the Non-negative Quadratic Programming (NQP):

$$\min_{\mathbf{v}} \frac{1}{2} \mathbf{v}^T \Phi \mathbf{v} + \mathbf{c}^T \mathbf{v} \quad \text{s.t. } 0 \leq \mathbf{v} \leq \mathbf{vmax}, \quad (12)$$

where the matrix Φ is assumed to be symmetric and positive defined, and \mathbf{vmax} is some positive constant. The multiplicative updates for the NQP are summarized as follows (see [17] for details on derivation and convergence):

$$\Phi_{nl}^+ = \begin{cases} \Phi_{nl} & \text{if } \Phi_{nl} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \Phi_{nl}^- = \begin{cases} |\Phi_{nl}| & \text{if } \Phi_{nl} < 0 \\ 0 & \text{otherwise} \end{cases},$$

$$\mathbf{v}^{(k+1)} = \min \left\{ \mathbf{v}^{(k)} \left[\frac{-\mathbf{c} + \sqrt{\mathbf{c}^2 + \mathbf{v}^{(k)} \mathbf{v}^{(k)}}}{2 \mathbf{v}^{(k)}} \right], \mathbf{vmax} \right\} \quad (13)$$

where $\mathbf{v}^{(k)} = \Phi^+ \mathbf{v}^{(k)}$, $\mathbf{v}^{(k)} = \Phi^- \mathbf{v}^{(k)}$ and all algebraic operations in (13) are to be carried out element wise. The NQP is quite efficient and has been used to solve interesting problems such as statistical learning [17], compressive sensing [21], etc. and in general, it is well-suited to solve large-scale non-negative quadratic programming problems, since its update rule only involves matrix-vector multiplication.

In [17] it is proven that the multiplicative updates (13) converge to the solution of (12), nevertheless there is no explicit mention of a practical rule to stop the iterations; in the next section we propose a simple but effective practical condition.

2.4 Vector-valued IRN-NQP

The optimization problem defined in (10) is unconstrained, and within the IRN framework ([18, 9]), the linear system (11) is usually solved via conjugate gradient (CG) or Preconditioned CG (PCG). By setting $\Phi^{(k)} = \tilde{A}^T W^{(k)} \tilde{A}$ and $\mathbf{c} = -\tilde{A}^T W^{(k)} \tilde{\mathbf{b}}$, in (12), a subtle but significant result is achieved: the constraint $0 \leq \mathbf{u} \leq \mathbf{vmax}$ is enforced; in other words, by using the proposed change of variables we may solve the non-negative constraint optimization problem described in (2) by iteratively solving (10) via the multiplicative updates described in (13). We stress that the upper bound constraint may or may not be enforced (see Sections 2.1 and 2.3 in [17]) but we consider it since it could be useful when a priori information about a physically meaningful upper bound is known.

The vector-valued IRN-NQP algorithm is summarized in Algorithm 1. We highlight the matrix $\Phi^{(k)}$ does not have to be explicitly constructed (neither $\Phi^{+(k)}$ nor $\Phi^{-(k)}$); for instance if Φ (we drop the superscript to simplify the notation) is equivalent to the convolution operation, i.e.: $\Phi \mathbf{u} = g * \mathbf{u}$, where g is a convolution kernel (and $*$ the convolution operator), then we define $g = g^+ - g^-$ (g^+ and g^- defined in a similar fashion as Φ^+ and Φ^- in (13)) to finally note that $\Phi^+ \mathbf{u} = g^+ * \mathbf{u}$ and $\Phi^- \mathbf{u} = g^- * \mathbf{u}$.

One key aspect, with a direct impact to the computational and reconstruction performance of the vector-valued IRN-NQP algorithm, is when to stop the multiplicative updates described in (13); in [17] there is no explicit mention about this important detail. Here we propose to use simple condition (see $\varepsilon_{NQP}^{(k)}$ in Algorithm 1, which we called *NQP tolerance*) to terminate the inner loop in Algorithm 1; this condition, inspired in the idea of forcing terms [22] for the Inexact Newton method [23], adapts the tolerance used to decide when to stop the multiplicative updates (break the inner loop). Experimentally, we have determined that the constants $\alpha \in [1 \dots 0.5]$ and $\gamma \in [1e-3 \dots 5e-1]$ give a good compromise between computational and reconstruction performance.

Initialize

$$\mathbf{u}^{(0)} = \mathbf{b}$$

for $k = 0, 1, \dots$

$$W_F^{(k)} = \text{diag} \left(\tau_{F, \varepsilon_F} (A \mathbf{u}^{(k)} - \mathbf{b}) \right)$$

$$\Omega_R^{(k)} = \text{diag} \left(\tau_{R, \varepsilon_R} \left(\sum_{n \in C} (D_x \mathbf{u}_n^{(k)})^2 + (D_y \mathbf{u}_n^{(k)})^2 \right) \right)$$

$$W_R^{(k)} = I_6 \otimes \Phi^{(k)}, \quad D = I_3 \otimes [D_x^T D_y^T]^T$$

$$\Phi^{(k)} = A^T W_F^{(k)} A + \lambda D^T W_R^{(k)} D$$

$$\mathbf{c}^{(k)} = -A^T W_F^{(k)} \mathbf{b}$$

$$\mathbf{u}^{(k,0)} = \mathbf{u}^{(k)}$$

$$\varepsilon_{NQP}^{(k)} = \gamma \cdot \left(\frac{\|\Phi^{(k)} \mathbf{u}^{(k,0)} + \mathbf{c}^{(k)}\|_2}{\|\mathbf{c}^{(k)}\|_2} \right)^\alpha \quad (\text{NQP tolerance})$$

for $m = 0, 1, \dots, M$

$$\mathbf{v}^{(k,m)} = \Phi^{+(k)} \mathbf{u}^{(k,m)}, \quad \mathbf{v}^{(k,m)} = \Phi^{-(k)} \mathbf{u}^{(k,m)}$$

$$\mathbf{u}^{(k,m+1)} = \min \left\{ \mathbf{u}^{(k,m)} \left[\frac{-\mathbf{c}^{(k)} + \sqrt{\mathbf{c}^{(k)2} + \mathbf{v}^{(k,m)} \mathbf{v}^{(k,m)}}}{2\mathbf{v}^{(k,m)}} \right], v_{max} \right\}$$

$$\text{if} \left(\frac{\|\Phi^{(k)} \mathbf{u}^{(k,m+1)} + \mathbf{c}^{(k)}\|_2}{\|\mathbf{c}^{(k)}\|_2} < \varepsilon_{NQP}^{(k)} \right) \text{break;}$$

end

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k,M+1)}$$

end

Algorithm 1: Vector-valued IRN-NQP algorithm.

Finally, we emphasize that the thresholds values for the weighting matrices W_F and W_R have a also great impact in the quality of the results and in the computational performance as well, and while not done so here, this algorithm can auto-adapt (as for the IRN algorithm, see [18, Sec. IV.G]) the

values of ε_F and ε_R based on the particular characteristics of the input data to be denoised/deconvolved.

3. EXPERIMENTAL RESULTS

To best of our knowledge, there is no other algorithm that explicitly includes a non-negativity constraint for color images within the TV framework, and therefore we choose to compare the vector-valued IRN-NQP (vv-IRN-NQP) algorithm with the vector-valued IRN (vv-IRN) algorithm [9], focusing on reconstruction quality, since the latter does not include a non-negativity constraint; however we do report the computational performance for both algorithms. It is important to highlight that in [9] the performance of the vv-IRN algorithm was compared with that of three alternative variational approaches: [5], an approximation of the *Mumford-Shah* functional, the vectorial lagged diffusivity (an extension of [24] used in [5]) and [25], an implementation of the fast dual minimization (FDM) of vector-valued TV [7]. For all cases the reconstruction quality were about the same, nevertheless the vv-IRN had a superior computational performance for the general deconvolution case; only the FDM method [7, 25] for ℓ^2 -VTV denoising had a slightly better computational results.

We used the color natural images from [26] as test images, which includes aerial, texture and miscellaneous images (including the standards “Lena” and “Goldhill”, see Fig. 1). The images are between 512×512 and 1024×1024 pixel. All simulations have been carried out using Matlab-only code on a 1.83GHz Intel Dual core CPU (L2: 2048K, RAM: 4G). Results corresponding to the vector-valued IRN-NQP algorithm presented here may be reproduced using the the NUMIPAD (v. 0.30) distribution [27], an implementation of IRN and related algorithms.



Figure 1: Test color images: (a) “Goldhill” (787×576 pixel), and (b) “Lena” (512×512 pixel).

All images were used for the denoising and deconvolution cases, where we corrupted the test images with Gaussian additive noise or salt-and-pepper noise. For the deconvolution case, the images were blurred by 7×7 out-of-focus kernel (2D pill-box filter). Due to space constraints we chose to present reconstruction SNR values and computation times for the “Goldhill” and “Lena” test images only; in average these results are representative of all test images. Reconstruction SNR values and computation times are compared in Table 1 (denoising with the salt-and-pepper noise model or ℓ^1 -VTV denoising), Table 2 (denoising with the Gaussian noise model or ℓ^2 -VTV denoising), Table 3 (deconvolution with the salt-and-pepper noise model or ℓ^1 -VTV deconvolution), Table 4 (deconvolution with the Gaussian noise model

or ℓ^2 -VTV deconvolution). In Figs. 2, and 3 we displayed noisy and reconstructed images for the ℓ^1 -VTV denoising, ℓ^1 -VTV deconvolution and ℓ^2 -VTV deconvolution respectively.



Figure 2: (a) “Goldhill” with 50% of its pixel corrupted with salt and pepper noise (SNR -4.81 dB) (b) “Goldhill” reconstructed via the vv-IRN-NQP algorithm (SNR 15.0 dB).

Image	Noise	SNR (dB)		Time (s)	
		vv-IRN	vv-IRN-NQP	vv-IRN	vv-IRN-NQP
Goldhill	10%	21.7	21.9	58.7	66.8
	30%	17.7	17.7	79.8	107.2
	50%	15.2	15.0	102.7	183.3
	70%	12.0	11.5	113.9	209.4
Lena	10%	22.5	23.1	37.7	47.4
	30%	18.6	19.0	48.1	73.5
	50%	16.1	16.5	53.5	134.7
	70%	12.5	12.6	63.5	142.8

Table 1: Denoising performance results for ℓ^1 -VTV, computed via the vv-IRN [9] and the vv-IRN-NQP algorithms.

SNR values are about the same for all cases with moderate noise levels, nevertheless the images reconstructed via the vv-IRN-NQP method have better visual quality. For high noise levels, we particularly point out the deconvolution performance (reconstruction quality) results for ℓ^1 -VTV case: the proposed method results has far better SNR values than the vv-IRN [9] method (see Table 3); moreover the reconstructed images via vv-IRN have negative pixel values, which produces an unpleasant visual effect, and setting them to zero may result in the presence of spurious ripples in the reconstructed image. The reconstructed images via vv-IRN-NQP (proposed method) does not suffer from these kind of artifacts due to the non-negativity constraint.

As expected the computational performance of the vv-IRN (which does not include a non-negativity constraint) is better than performance of our proposed method (vv-IRN-NQP algorithm), especially for the deconvolution case (about 9 times faster). Experimentally, we have found that the vv-IRN-NQP method needs high accuracy (we use $\alpha = 0.5$, $\gamma = 1e-3$ in Algorithm 1) to attain good reconstruction results for the deconvolution case; one way to improve the computational performance of the vv-IRN-NQP method could be to seed it with a *good* initial solution (found via any unconstrained VTV algorithm) so only a few iterations are needed.



Figure 3: Blurred (a) “Lena” with 50% of its pixel corrupted with salt and pepper noise (SNR -4.55 dB) (b) “Lena” reconstructed via the vv-IRN-NQP algorithm (SNR 17.9 dB).

4. CONCLUSIONS

The vv-IRN-NQP algorithm gives very good reconstruction quality for the ℓ^2 -VTV and ℓ^1 -VTV denoising/deconvolution problems (specially for very high levels of noise), with a satisfactory computational performance, even when compared to the vv-IRN method [9]. However, to the best of our knowledge, the vv-IRN-NQP method is the only algorithm that explicitly includes a non-negativity constraint for color images within the TV framework, and it was expected that its computational performance wouldn’t be as good as those methods which do not include a non-negativity constraint.

Finally, we highlight that the vv-IRN-NQP algorithm is very flexible, and some of its parameters, such the tolerance to stop the multiplicative updates and thresholds for weighting matrices, can be automatically adapted to the particular input dataset. Furthermore, it 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 ℓ^1 -TV problems.

Image	Noise (σ^2)	SNR (dB)		Time (s)	
		vv-IRN	vv-IRN-NQP	vv-IRN	vv-IRN-NQP
Goldhill	2.5e-3	19.0	18.9	15.2	27.1
	1.0e-2	16.2	16.2	18.7	27.4
Lena	2.5e-3	19.9	19.9	9.7	15.1
	1.0e-2	17.2	17.2	11.3	15.7

Table 2: Denoising performance results for ℓ^2 -VTV, computed via the vv-IRN [9] and the vv-IRN-NQP algorithms.

REFERENCES

- [1] P. Rodríguez, “Multiplicative updates algorithm to minimize the generalized total variation functional with a non-negativity constraint,” Submitted to IEEE International Conference on Image Processing (ICIP), 2010.
- [2] T. Chan, S.Kang, and J. Shen, “Total variation denoising and enhancement of color images based on the CB and HSV color models,” *J. Visual Comm. Image Rep.*, vol. 12, pp. 422–435, 2001.
- [3] J. Weickert and T. Brox, “Diffusion and regularization of vector and matrix-valued images,” Tech. Rep., Universität des Saarlandes, 2002.

Image	Noise	SNR (dB)		Time (s)	
		vv-IRN	vv-IRN-NQP	vv-IRN	vv-IRN-NQP
Goldhill	10%	20.2	19.8	151.1	1278.9
	30%	19.2(*)	18.0	207.4	1292.2
	50%	13.7(*)	16.5	186.8	1275.7
	70%	6.2(*)	12.9	104.5	1276.1
Lena	10%	21.3	21.3	95.9	845.1
	30%	20.2(*)	19.4	130.8	842.6
	50%	14.7(*)	17.9	91.3	883.9
	70%	6.6(*)	13.9	65.9	873.4

Table 3: Deconvolution performance results for ℓ^1 -VTV, computed via the vv-IRN [9] and the vv-IRN-NQP algorithms. Reconstructed images with SNR values marked with (*) have negative pixel values.

- [4] D. Tschumperle and R. Deriche, "Vector-valued image regularization with pdes: A common framework for different applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, 2005.
- [5] L. Bar, A. Brook, N. Sochen, and N. Kiryati, "Deblurring of color images corrupted by impulsive noise," *IEEE TIP*, vol. 16, pp. 1101–1111, 2007.
- [6] O. Christiansen, T. Lee, J. Lie, U. Sinha, and T. Chan, "Total variation regularization of matrix-valued images," in *Int. J. Biomed Imaging*, 2007.
- [7] X. Bresson and T. Chan, "Fast dual minimization of the vectorial total variation norm and applications to color image processing," *J. of Inverse Problems and Imaging*, vol. 2, no. 4, pp. 455–484, 2008.
- [8] Y. Wen, M. Ng, and Y. Huang, "Efficient total variation minimization methods for color image restoration," *IEEE TIP*, vol. 17, no. 11, pp. 2081–2088, 2008.
- [9] P. Rodríguez and B. Wohlberg, "A generalized vector-valued total variation algorithm," in *Proceedings of the International Conference on Image Processing (ICIP)*, Nov. 2009.
- [10] P. Blomgren and T. Chan, "Color TV: Total variation methods for restoration of vector valued images," *IEEE TIP*, vol. 7, no. 3, pp. 304–309, 1998.
- [11] A. Bonnet, "On the regularity of edges in image segmentation," *Annales de l'institut Henri Poincaré (C) Analyse non linéaire*, vol. 13, no. 4, pp. 485–528, 1996.
- [12] N. Galatsanos, A. Katsaggelos, T. Chin, and A. Hillery, "Least squares restoration of multichannel images," *IEEE TSP*, vol. 39, no. 10, pp. 2222–2236, 1991.
- [13] M.R. Banham, N.P. Galatsanos, H.L. Gonzalez, and A.K. Katsaggelos, "Multichannel restoration of single channel images using a wavelet-based subband decomposition," *IEEE TIP*, vol. 3, no. 6, pp. 821–833, November 1994.
- [14] D. Krishnan, P. Lin, and A. Yip, "A primal-dual active-set method for non-negativity constrained total variation deblurring problems," *IEEE T. on Image Processing*, vol. 16, no. 11, pp. 2766–2777, 2007.
- [15] C. Vogel, *Computational Methods for Inverse Problems*, SIAM, 2002.
- [16] R. Chartrand and B. Wohlberg, "Total-variation regularization with bound constraints," Accepted for presentation at the 2010 IEEE ICASSP, 2010.
- [17] F. Sha, Y. Lin, L. Saul, and D. Lee, "Multiplicative updates for nonnegative quadratic programming," *Neural Comput.*, vol. 19, no. 8, pp. 2004–2031, 2007.
- [18] P. Rodríguez and B. Wohlberg, "Efficient minimization method for a generalized total variation functional," *IEEE TIP*, vol. 18, no. 2, pp. 322–332, 2009.
- [19] A. Beaton and J. Tukey, "The fitting of power series, meaning polynomials illustrated on band-spectroscopic data," *Technometrics*, , no. 16, pp. 147–185, 1974.
- [20] J. Scales and A. Gersztenkorn, "Robust methods in inverse theory," *Inverse Problems*, vol. 4, no. 4, pp. 1071–1091, Oct. 1988.
- [21] P. O'Grady and S. Rickard, "Recovery of non-negative signals from compressively sampled observations via non-negative quadratic programming," in *SPARS'09*, Saint Malo, France, Apr. 2009.
- [22] S. Eisenstat and H. Walker, "Choosing the forcing terms in an inexact Newton method," *SIAM J. Sci. Comp.*, vol. 17, no. 1, pp. 16–32, 1996.
- [23] Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug, "Inexact Newton methods," *SIAM J. Sci. Comp.*, vol. 19, no. 2, pp. 400–408, 1982.
- [24] C. Vogel and M. Oman, "Iterative methods for total variation denoising," *SIAM J. on Sci. Comp.*, vol. 17, no. 1, pp. 227–238, 1996.
- [25] P. Getreuer, "Total variation grayscale and color image denoising," Matlab central file <http://www.mathworks.com/matlabcentral/fileexchange/16236>.
- [26] Signal and I. P. I. of the University of Southern California, "The USC-SIPI image database," <http://sipi.usc.edu/database/>.
- [27] P. Rodríguez and B. Wohlberg, "Numerical methods for inverse problems and adaptive decomposition (NUMIPAD)," <http://numipad.sourceforge.net/>.

Image	Noise (σ^2)	SNR (dB)		Time (s)	
		vv-IRN	vv-IRN-NQP	vv-IRN	vv-IRN-NQP
Goldhill	2.5e-3	15.2	15.2	35.4	213.6
	1.0e-2	14.1	14.1	70.9	210.0
Lena	2.5e-3	16.4(*)	16.5	22.5	105.9
	1.0e-2	15.3	15.3	42.6	116.4

Table 4: Deconvolution performance results for ℓ^2 -VTV, computed via the vv-IRN [9] and the vv-IRN-NQP algorithms. Reconstructed images with SNR values marked with (*) have negative pixel values.