Digital Signal Processing 40 (2015) 101–116

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

A total variation recursive space-variant filter for image denoising

Manya V. Afonso *, João M.R. Sanches

Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisbon, Portugal

ARTICLE INFO

Article history:
Available online 20 February 2015

Keywords:
Denoising
Inpainting
Total variation
Non-Gaussian noise
Space variant filtering

ABSTRACT

Total Variation (TV) regularization is a widely used convex but non-smooth regularizer in image restoration and reconstruction. Many algorithms involve solving a denoising problem as an intermediate step or in each iteration. Most existing solvers were proposed in the context of a specific application. In this paper, we propose a denoising method which can be used as a proximal mapping (denoising operator) for noises other than additive and Gaussian. We formulate the Maximum A-Posteriori (MAP) estimation in terms of a spatially adaptive and recursive filtering operation on the Maximum Likelihood (ML) estimate. The only dependence on the model is the ML estimate and the second order derivative, which are computed at the beginning and remain fixed throughout the iterative process. The proposed method generalizes the MAP estimation with a quadratic regularizer using an infinite impulse response filter, to the case with TV regularization. Due to the fact that TV is non-smooth and has spatial dependencies, the resulting filter after reweighted least squares formulation of the TV term, is recursive and spatially variant. The proposed method is an instance of the Majorization–Minimization (MM) algorithms, for which convergence conditions are defined and can be shown to be satisfied by the proposed method. The method can also be extended to image inpainting and higher order TV in an intuitively straight-forward manner.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Image denoising/reconstruction involves estimating an image or three dimensional (3D) volume from a set of noisy and often incomplete set of observations. The relationship between the image to be estimated, $\mathbf{x}$ and the observation, $\mathbf{y}$, and the type of noise corrupting the observed image depends on the imaging modality. For example, thermal noise in imaging sensors is usually modeled as additive and Gaussian distributed, multiplicative speckle noise (which occurs in Ultrasound and Synthetic Aperture Radar imaging) is widely assumed to be Rayleigh distributed [51], and Poisson noise in confocal fluorescence microscopy [43]. While denoising for the case of additive and Gaussian noise has received a lot of interest because of its mathematical tractability, it is not always applicable in several situations [29].

Estimating the image is an ill-posed problem, necessitating prior information. In the Bayesian framework [58], we assume a suitable prior density on $\mathbf{x}$, $p(\mathbf{x})$, construct the posterior density $p(\mathbf{x}|\mathbf{y})$, and compute the Maximum A-posteriori Probability (MAP) estimate of $\mathbf{x}$ as the value which maximizes this probability. The prior or regularizer must be chosen so as to make the optimization process feasible and numerically stable, and enforce certain characteristics on the estimate. The well known convex optimization formulation for estimating $\mathbf{x}$ given $\mathbf{y}$ involves solving the minimization problem,

$$\min_{\mathbf{x}} l(\mathbf{x},\mathbf{y}) + \frac{\lambda}{2}\phi(\mathbf{x}), \tag{1}$$

where $l(\mathbf{x},\mathbf{y})$ is the data fidelity term, and is convex, $\phi(.)$ is the regularizer function, and $\lambda > 0$ is the parameter that controls the influence of the regularizer. It is trivial to show that the solution to this problem is the Bayesian Maximum A-posteriori (MAP) estimate of $\mathbf{x}$, where $l(\mathbf{x},\mathbf{y})$ is the negative log likelihood function, and the prior on $\mathbf{x}$ is proportional to $e^{-\lambda\phi(\mathbf{x})}$. The data fidelity function depends on the statistics of the observation model. Refer to Table 1 for a summary of the likelihood density, log likelihood ratio, and the maximum likelihood (ML) estimate for the additive and Gaussian, multiplicative and Rayleigh, and Poisson noise models.

An algebraically elegant and computationally simple method for obtaining the MAP estimate with a quadratic prior was presented in [48]. In this work, a Taylor series approximation was used to formulate the data fidelity term in terms of the ML estimate, which led to a closed form minimizer of (1) which could be implemented as a cascade of two Infinite Impulse Response (IIR) filters. The filter coefficients and other parameters were dependent on the model

* Corresponding author.
E-mail addresses: mafonso@isr.ist.utl.pt (M.V. Afonso), jmrs@isr.ist.utl.pt (J.M.R. Sanches).

http://dx.doi.org/10.1016/j.dsp.2015.02.002
1051-2004/© 2015 Elsevier Inc. All rights reserved.

Table 1
Parameters for the Gaussian, Rayleigh and Poisson statistical models.

	Gaussian	Rayleigh	Poisson
Likelihood $p(\mathbf{y}\|\mathbf{x})$	$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_i (y_i - x_i)^2}$	$\prod_i \frac{y_i}{x_i} e^{-\frac{y_i^2}{2x_i}}$	$\prod_i \frac{x_i^{y_i} e^{-x_i}}{(y_i!)}$
Data fidelity $l(\mathbf{x}, \mathbf{y})$	$\frac{1}{2\sigma^2}\sum_i (y_i - x_i)^2$	$\sum_i \left[\frac{y_i^2}{2x_i} - \log(\frac{y_i}{x_i})\right]$	$\sum_i [x_i - y_i \log(x_i)]$
$\gamma_i = \partial^2 l / \partial x_i^2$	$\frac{1}{\sigma^2}$	$\frac{1}{(x_i^{ML})^2}$	$\frac{1}{x_i^{ML}}$
ML estimate x^{ML}	y	$\frac{1}{2}y^2$	y

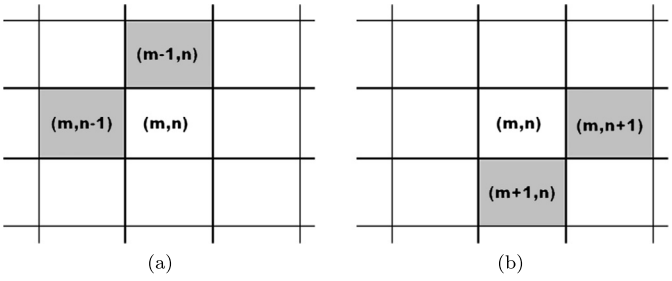

Fig. 1. Neighborhoods used to assign filter coefficients. (a) Neighborhood $\mathcal{N}_i$ over which the weight $w_{(m,n)}$ is applied, shown as the shaded pixels. (b) Neighborhood $\mathcal{N}_i'$ in which the pixel $i = (m, n)$ appears in the difference terms.

and computed beforehand. This framework has the advantage of being computationally simple and applicable to different statistical models. In this work, our goal is to extend this MAP estimation filtering framework to solve the denoising problem with the total variation (TV) [11,45] regularizer which enforces the desirable qualities of sparseness and piece-wise smoothness [55].

The TV semi-norm is convex but non-smooth, and has led to the development of several image denoising, deconvolution, and reconstruction methods [10,56]. In denoising natural images without texture, under the additive and Gaussian distributed noise model it is considered state-of-the-art in terms of Improvement in Signal to Noise Ratio (ISNR) [6]. In our notation, $\mathbf{x}$ is the vector representation, say, in lexicographic ordering, of a two-dimensional image and an index coordinate i corresponds to a pixel location (m, n). We will use the vector representation throughout. However, because of the fact that the two first-difference neighbors of a pixel $x_{(m,n)}$, $x_{(m-1,n)}$ and $x_{(m,n+1)}$ cannot be represented contiguously in this representation, we will use indices i and (m, n) interchangeably.

The isotropic TV semi-norm of a two-dimensional (2D) image $\mathbf{x}$ indexed by coordinates (m, n), assuming suitable boundary conditions, is defined as,

$$\mathrm{TV}(\mathbf{x}) = \sum_{(m,n)} \sqrt{(x_{m,n} - x_{m-1,n})^2 + (x_{m,n} - x_{m,n-1})^2}. \tag{2}$$

A generalized expression valid for higher dimensions is,

$$\mathrm{TV}(\mathbf{x}) = \sum_i \sqrt{\sum_{\mathbf{j}\in\mathcal{N}_\mathbf{i}} (x_i - x_\mathbf{j})^2}, \tag{3}$$

where the index i (which in the vector representation corresponds to the coordinates (m, n)) indexes the image or volume, and $\mathcal{N}_i$ is its neighborhood (see Fig. 1), over which the difference is computed. Unless stated otherwise, the neighborhood $\mathcal{N}_i^C$ is the first order neighborhood. For the 2D case the index is $i \widehat{=} (m, n)$ and $\mathcal{N}_i^C = \{x_{(m-1,n)}, x_{(m,n-1)}, x_{(m+1,n)}, x_{(m,n+1)}\}$.

1.1. Related work

Several solvers exist for the case when the noise is additive and Gaussian [10,56,27], leading to the TV-regularized least squares minimization also known as the TV-ℓ_2 problem,

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + \frac{\lambda}{2}\mathrm{TV}(\mathbf{x}). \tag{4}$$

The separability of the ℓ_2 term has been advantageous in the development of solvers. The Chambolle method [10] involves iteratively calculating a projection onto a convex set. More recent methods based on the Bregman iterations and Alternating Directions Method of Multipliers (ADMM) frameworks [56,27] have benefited from the separability of the data fidelity term, combined with variable splitting [16] to have a simpler least squares minimization step at each iteration.

When the noise is not additive and Gaussian, $l(\mathbf{x}, \mathbf{y})$ is no longer a least squares term and problem (4) cannot be used to estimate $\mathbf{x}$. There has not been much literature on general solvers for TV denoising with noise other than additive and Gaussian. Even though TV has been used in denoising in the context of multiplicative despeckling with Rayleigh or Gamma distributed noise [52,7], and Poisson noise [43], these methods are application specific and are not general solvers for their respective noise models. In [12], a TV regularized denoising method for Poisson and impulse noise was proposed, which used the quasi-Newton method to minimize the sum of the TV term with the appropriate data fidelity term. A review of TV based denoising and restoration methods for Gaussian, gamma-speckle, Poisson, and impulse noises can be found in [44].

In [47,48], the authors use a Taylor series approximation around the ML estimate to have a weighted least squares data fidelity term, irrespective of the observation model. With a quadratic Gibbs regularizer, the MAP estimate had a closed form, which can be solved using two IIR filters in sequence. This is possible because the quadratic and Tikhonov [28] regularizers are convex and differentiable.

Yet another family of algorithms for denoising and reconstruction is the Iterative Reweighted Least Squares (IRLS) type algorithms. The IRLS method has been around for half a century [34,40,35], but has been applied in sparse signal recovery for a little over a decade [30,42,20,13,14,17] with the more recent works focusing on reconstruction [17] from compressive sensing [9,19] measurements. These methods, all proposed in the context of additive and Gaussian noise, solve denoising problems with the ℓ_1-norm or non-convex ℓ_p-norm $(0 < p < 1)$ regularizers which are non-smooth and therefore not easily differentiable. Therefore, the regularizer term is formulated as a weighted ℓ_2 term, and the least squares problem is solved iteratively, with the weights updated at each iteration. We will use the IRLS formulation to reformulate the TV term as a weighted sum of squared differences.

Non-TV based image denoising methods include those based on wavelet type representations such as BM3D [15], principal component analysis (PCA) [36,46], and those based on the non-local means (NL-means) algorithm [41]. Recent work such as [3,60] are

based on neural networks. In these approaches, denoising and inpainting problems are solved using deep networks trained with sparse denoising auto-encoders (SDAs) to learn a dictionary, followed by sparse minimization with a sparsity inducing regularizer such as the Frobenius norm. A median filter based method for denoising non-Gaussian and power law distributed noise was proposed in [26], consisting of a thresholding operation, followed by median filtering, followed by BM3D filtering. In our proposed filtering approach, we do not learn a set of bases but iteratively and adaptively adjust the weights corresponding to the reweighted TV formulation.

1.2. Contributions

In this work, we combine the IRLS method with the MAP estimation using IIR filtering, to propose a filtering approach for TV denoising, for the Gaussian, Rayleigh, and Poisson observation models. As in [47,48], we use a Taylor series approximation around the ML estimate to always have a weighted quadratic data fidelity term instead of the one for the corresponding model. In case of the TV regularizer, the problem is complicated by the fact that the regularizer term is not separable but based on spatial differences between neighboring pixels.

We formulate the TV term as a weighted sum of the squares of the differences between neighboring pixels. We therefore need to update the weights at each iteration as in IRLS. Because of the spatial dependencies, the filter is now not only recursive but also spatially adaptive. Only the ML estimate and other filter parameters depend on the noise model, and therefore the proposed approach can be used as a TV denoising solver for the respective model. We show that the proposed filter is an instance of the Majorization–Minimization (MM) family of algorithms [32,24,39], for which convergence has been proved [54,59].

A simple and intuitive modification to the filtering operation allows the method to be extended to the problem of image inpainting [37,5,53], when the image needs to be estimated with some pixels missing from the noisy observed image.

The proposed adaptive filtering framework can be easily extended to TV terms taking into account higher order differences [33].

Experimental results show that the proposed method achieves a performance comparable to state-of-the-art methods for Gaussian, Rayleigh, and Poisson noise.

1.3. Organization of the paper

In this section we have motivated the problem and briefly summarized the different and not directly related concepts we make use of, namely, Bayesian MAP inference through filtering, Taylor series approximation to generalize the convex problem, and IRLS. We present the proposed approach in Section 2, and extend it to higher order TV and image inpainting in Section 3. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. Proposed method

Our starting point is (1). For clarity, we reformulate the convex problem to be solved to obtain the MAP estimate as the minimizer of the energy function $E(\mathbf{y},\mathbf{x})$ which is the sum of the data fidelity term and the regularizer,

$$\mathbf{x}^{MAP} = \arg\min_{\mathbf{x}} E(\mathbf{y},\mathbf{x}), \tag{5}$$

$$= \arg\min_{\mathbf{x}} l(\mathbf{y},\mathbf{x}) + \frac{\lambda}{2} TV(\mathbf{x}). \tag{6}$$

In [48,47], a computationally simple and elegant approach is presented that transforms a non-Gaussian data fidelity term into a least squares function in terms of the ML estimate. It is assumed that the data fidelity term $l(\mathbf{y},\mathbf{x})$ is separable, and we can solve for each element x_i of $\mathbf{x}$, using a Gauss–Seidel approach. Note that separability holds for denoising problems, but not for reconstruction or deblurring where there can be interactions between pixels. It is also assumed that the difference between the ML and MAP estimates is small. Each pixel x_i is estimated by solving,

$$\frac{\partial E(\mathbf{y},\mathbf{x})}{\partial x_i} = 0. \tag{7}$$

Approximating the likelihood function by the second order Taylor series, computed at the ML estimate, $\mathbf{x}^{ML}$, leads to,

$$l(\mathbf{y},\mathbf{x}) \simeq l(\mathbf{y},\mathbf{x}^{ML}) + \frac{\partial l}{\partial x_i}(\mathbf{y},\mathbf{x}^{ML})(x_i - x_i^{ML}) + \frac{1}{2}\frac{\partial^2 l}{\partial x_i^2}(\mathbf{y},\mathbf{x}^{ML})(x_i - x_i^{ML})^2 \tag{8}$$

In (8), independently of the statistical model, the data fidelity term is expressed as a quadratic polynomial. The first two terms can be discarded, because after differentiation w.r.t. x_i, they yield no term in x_i. The term $\frac{\partial l}{\partial x_i}(\mathbf{y},\mathbf{x}^{ML})$ is zero by definition, because the ML estimate x_i^{ML} is a stationary point of the likelihood function. The parameter $\gamma_i = \frac{\partial^2 l}{\partial x_i^2}(\mathbf{y},\mathbf{x}^{ML})$ depends on the model and the ML estimate, but can be precomputed. The values for the Gaussian, Rayleigh, and Poisson models are presented in Table 1.

Our Taylor series reformulated problem with TV regularization is now,

$$\mathbf{x}^{MAP} = \arg\min_{\mathbf{x}} \sum_i \gamma_i (x_i - x_i^{ML})^2 + \frac{\lambda}{2} TV(\mathbf{x}). \tag{9}$$

In [48], a quadratic Gibbs prior $p(\mathbf{x}) = \exp(-\sum_i |x_i - x_{i-1}|^2)$ was used leading to the optimization problem,

$$\mathbf{x}^{MAP} = \arg\min_{\mathbf{x}} \sum_i \gamma_i (x_i - x_i^{ML})^2 + \lambda \sum_i (x_i - x_{i-1})^2. \tag{10}$$

To solve for a pixel x_i from the MAP estimate, the objective function from (10) is differentiated w.r.t. x_i, the derivative is equated to zero, and the resulting linear equation is solved as,

$$2\gamma_i (x_i - x_i^{ML}) + 2\lambda (x_i - x_{i-1}) + 2\lambda (x_i - x_{i+1}) = 0, \tag{11}$$

which leads to,

$$x_i = \frac{\gamma_i}{\gamma_i + 2\lambda} x_i^{ML} + \frac{\lambda}{\gamma_i + 2\lambda}(x_{i-1} + x_{i+1}). \tag{12}$$

This expression can be implemented as an IIR filter, with the ML estimate as the input. Since (12) has a causal and anti-causal part, in [48], it was implemented as a cascade of two first order filters. First the ML estimate was filtered by the causal component, then the result was reversed in time and then filtered by the anti-causal component (implemented as a causal filter, after time reversal). Thus the MAP estimate with a quadratic or Tikhonov [28] prior was obtained as a result of two first order filtering operations.

Obviously, when we replace the regularizer term from (11) with an ℓ_p-norm ($0 < p \leq 1$) or an absolute difference term $\sum_i |x_i - x_{i-1}|$, we cannot perform the MAP estimation applying two filters because of non-smoothness. For these non-smooth regularizers, a reformulation must be applied to express the non-smooth term as a weighted quadratic term, and iteratively solve the minimization problem and adjust the weights. This is the rationale behind the IRLS algorithm for ℓ_p-norm ($0 < p \leq 1$) regularization.

For an element $x_i \in \mathbb{R}$, the absolute value is expressed as $|x_i| = w_i x_i^2$, where the weight $w_i = 1/|x_i|$. Thus, the ℓ_1-norm of $\mathbf{x}$ can be expressed as the weighted ℓ_2-norm,

$$\|x\|_1 = \sum_i w_i x_i^2 = \mathbf{x}^T \mathbf{W} \mathbf{x},$$

where $\mathbf{W}$ is a diagonal matrix containing the weight w_i along its diagonal. It has been shown that IRLS with the pth power of the ℓ_p-norm when $0 < p < 1$ yields the solution with the non-convex ℓ_p norm [31]. To prevent the weights from becoming very large when the pixel values are very small, a real value $\epsilon > 0$ is added to the denominator $w_i = 1/(|x_i| + \epsilon)$. The weights can therefore be considered to be regularized [17]. The generalized IRLS (GIRLS) algorithm [38] reformulates the sum of the absolute differences between adjacent elements of a 1D signal, as a weighted quadratic term. For a pair of pixels, we have $|x_i - x_{i-1}| = w_i (x_i - x_{i-1})^2$, with $w_i = 1/(|x_i - x_{i-1}| + \epsilon)$.

2.1. Recursive and spatially adaptive filtering for TV

We can now see that having gone from a quadratic Gibbs prior to a non-smooth ℓ_p regularizer ($0 < p \leq 1$), we need to run a few iterations recursively to update the weights, as opposed to (12), which can be implemented as two filtering operations in cascade. The TV regularizer involves differences with neighboring pixels, and thus the filtering operation is not only recursive, but also spatially adaptive as will be shown.

For a pixel indexed by $i = (m, n)$, we define the neighborhoods over which the filtering is performed. In 2D, from the way (2) is computed, the neighborhood of first order differences is

$$\mathcal{N}_i = \{x_{(m-1,n)}, x_{(m,n-1)}\},$$

as shown in Fig. 1(a). Similarly, we define the neighborhood of pixels $\mathcal{N}_i'$, shown in Fig. 1(b), for which pixel i appears in the difference term and therefore is weighted by the coefficients corresponding to these pixels,

$$\mathcal{N}_i' = \{x_{(m+1,n)}, x_{(m,n+1)}\}.$$

Recall that the complete neighborhood around pixel i is $\mathcal{N}_i^C = \mathcal{N}_i \cup \mathcal{N}_i'$.

To apply the method to 3D or higher order differences, it is straightforward to define these neighborhoods.

As in [39,13], we express the TV term in terms of spatially adaptive filter coefficients as follows,

$$\begin{aligned} TV(\mathbf{x}) &= \sum \sqrt{\sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} (x_i - x_\mathbf{j})^2} \\ &= \sum_i w_i \left(\sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} (x_i - x_\mathbf{j})^2 \right) \end{aligned} \tag{13}$$

where the coefficient at location $\mathbf{i}$ is,

$$w_i = \frac{1}{\sqrt{\sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} (x_i - x_\mathbf{j})^2 + \epsilon}}. \tag{14}$$

We now substitute the TV term in (9), with the weighted quadratic formulation from (14) to obtain,

$$\min_{\mathbf{x}} \sum_i \gamma_i (x_i - x_i^{ML})^2 + \frac{\lambda}{2} \sum_i w_i \left(\sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} (x_i - x_\mathbf{j})^2 \right). \tag{15}$$

We can now solve element-wise by differentiating with respect to each x_i after gathering the terms in which it appears,

$$\begin{aligned} &\gamma_i (x_i - x_i^{ML}) \\ &\quad + 2\lambda w_i \sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} (x_i - x_\mathbf{j}) \\ &\quad + 2\lambda \sum_{\mathbf{k} \in \mathcal{N}_\mathbf{i}'} w_\mathbf{k} (x_i - x_\mathbf{k}) = 0. \end{aligned} \tag{16}$$

Hence, we have two first-order difference terms in (16), corresponding to the weight w_i of the pixel i, and to the sum of difference terms in which pixel i appears.

Rearranging the terms we get,

$$\begin{aligned} &(\gamma_i + 2\lambda w_i n_b + 2\lambda \sum_{\mathbf{k} \in \mathcal{N}_\mathbf{i}'} w_\mathbf{k}) x_i \\ &\quad = \gamma_i x_i^{ML} + 2\lambda w_i \sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} x_\mathbf{j} + 2\lambda \sum_{\mathbf{k} \in \mathcal{N}_\mathbf{i}'} w_\mathbf{k} x_\mathbf{k} \end{aligned} \tag{17}$$

where n_b is the size of the neighborhood. Thus the MAP estimate of x_i is the result of a spatially adaptive filter applied to the ML estimate,

$$x_i^{MAP} = a_i x_i^{ML} + b_i \sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} x_\mathbf{j} + c_i \sum_{\mathbf{k} \in \mathcal{N}_\mathbf{i}'} w_\mathbf{k} x_\mathbf{k}, \tag{18}$$

where $a_i = \gamma_i / d_i$, $b_i = 2\lambda w_i / d_i$, $c_i = 2\lambda / d_i$, with the denominator term $d_i = (\gamma_i + 2\lambda w_i n_b + 2\lambda \sum_{\mathbf{k} \in \mathcal{N}_\mathbf{i}'} w_\mathbf{k})$. However, this filter does not have a simple closed form because the weights w_i at a pixel $\mathbf{i}$ and the filter coefficients a_i, b_i, c_i are functions of the MAP estimate $\mathbf{x}$. Therefore, we need to use a recursive approach to estimate $\mathbf{x}$ and update these coefficients.

Eq. (18) can be expressed in matrix notation as,

$$\mathbf{x} = \mathbf{A}(\mathbf{x}) \mathbf{x}^{ML} + (\mathbf{B}(\mathbf{x}) + \mathbf{C}(\mathbf{x}))\, \mathbf{x}, \tag{19}$$

where row i of the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ correspond to the coefficients for pixel x_i. These matrices are expressed as functions of the MAP estimate. Gathering the terms in $\mathbf{x}$, the filter can be expressed in terms of a filtering operation on $\mathbf{x}^{ML}$ as follows,

$$\begin{aligned} &(\mathbf{I} - \mathbf{B}(\mathbf{x}) - \mathbf{C}(\mathbf{x}))\, \mathbf{x} = \mathbf{A}(\mathbf{x}) \mathbf{x}^{ML}, \\ &\mathbf{x} = (\mathbf{I} - \mathbf{B}(\mathbf{x}) - \mathbf{C}(\mathbf{x}))^{-1} \mathbf{A}(\mathbf{x}) \mathbf{x}^{ML}. \end{aligned} \tag{20}$$

Since the filter $(\mathbf{I} - \mathbf{B}(\mathbf{x}) - \mathbf{C}(\mathbf{x}))^{-1} \mathbf{A}(\mathbf{x})$ is a function of the MAP estimate $\mathbf{x}$, we need to use a recursive approach to estimate $\mathbf{x}$ and update the matrices $\mathbf{A}(\mathbf{x}), \mathbf{B}(\mathbf{x}), \mathbf{C}(\mathbf{x})$ at each iteration. This step is also solved recursively in [39] and [13], but with an approximate inversion, because the matrix $(\mathbf{I} - \mathbf{B}(\mathbf{x}) - \mathbf{C}(\mathbf{x}))$ is not easy to invert.

From (18), we express the estimate at iteration $t+1$ in terms of the recursive filter,

$$x_i^{t+1} = a_i(\mathbf{x}^t) x_i^{ML} + b_i(\mathbf{x}^t) \sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} x_\mathbf{j}^t + c_i(\mathbf{x}^t) \sum_{\mathbf{k} \in \mathcal{N}_\mathbf{i}'} w_\mathbf{k} x_\mathbf{k}^t. \tag{21}$$

The recursive, spatially varying linear filter representation at iteration $t+1$ is,

$$\mathbf{x}^{t+1} = A(\mathbf{x}^t) \mathbf{x}^{ML} + \left(B(\mathbf{x}^t) + C(\mathbf{x}^t) \right) \mathbf{x}^t. \tag{22}$$

This filtering process must be run for a few iterations, until the relative difference between coefficients $\{w_i\}^t$ and $\{w_i\}^{t+1}$ falls below a given threshold. From (13), we can expect that in homogeneous regions of the image, as the TV of the image decreases, because of the inverse relation, the amplitude of the coefficients $\{w_i\}$ would increase. In regions with texture, we would expect the TV to increase and the weights to decrease.

Our proposed method, which we call Adaptive Total Variation Filtering (ATVF) is summarized below:

Algorithm *ATVF*

1. Given the observation model and observation $\mathbf{y}$, compute $\mathbf{x}^{ML}$ and set of coefficients $\{\gamma_i\}$ using Table 1.
2. Initialize parameters $\lambda > 0$, $\epsilon > 0$, initial estimate $\mathbf{x}^{(0)}$.
3. Compute initial weights $\mathbf{w}^{(0)}$ using (14).
4. Set $t = 0$
5. **repeat**
6. Compute $\mathbf{x}^{(t+1)}$ using (22).
7. Update weights $\mathbf{w}^{(t+1)}$ with $\mathbf{x}^{(t+1)}$ using (14).
8. $t \leftarrow t + 1$
9. **until** stopping criterion is satisfied.

2.2. ATVF with regularized weights

At each iteration of ATVF, we need to compute (21) for each pixel. Upon inspecting (18) we find that the computational load comes from looking up the weight vectors corresponding to the pixel, for the vector inner product with the vector of neighboring pixels.

Notice that in the second term of (18) has a single weight w_i that is multiplied to the neighboring pixels from the set $\mathcal{N}_i$. We similarly approximate the third term, by assuming a form of regularization on the weight vector so as to multiply the pixels from the neighborhood $\mathcal{N}_i'$ by w_i. Eq. (18) now changes to,

$$x_i^{MAP} = a_i x_i^{ML} + b_i \sum_{\mathbf{k} \in (\mathcal{N}_\mathbf{i} \cup \mathcal{N}_\mathbf{i}')} x_\mathbf{k}, \tag{23}$$

where $a_i = \gamma_i / d_i$, $b_i = 4\lambda w_i / d_i$, with the denominator term $d_i = (\gamma_i + 4\lambda w_i)$.

We see that (adaptivefilterClosedFormRW) has no vector inner products because of the approximated weights and is therefore computationally simpler than (18). Using this equation for the first step in the iterative process in algorithm *ATVF* leads to an approximate method which we will refer to as ATVFRW for ATVF with Regularized Weights.

2.3. Convergence

From (9), the cost function to be minimized is,

$$L(\mathbf{x}) = \frac{1}{2} \sum_i \gamma_i (x_i - x_i^{ML})^2 + \frac{\lambda}{2} TV(\mathbf{x}). \tag{24}$$

Our reweighted problem (16) can be expressed in terms of a difference matrix $\mathbf{D}$, as

$$\min_\mathbf{x} \frac{1}{2} \sum_i \gamma_i (x_i - x_i^{ML})^2 + \frac{\lambda}{2} \mathbf{x}^T \mathbf{D}^T \mathbf{W} \mathbf{D} \mathbf{x}, \tag{25}$$

The difference operator is now a concatenation of the horizontal and vertical difference operators, $\mathbf{D}_h$ and $\mathbf{D}_v$ respectively, $\mathbf{D}^T = [\mathbf{D}_h^T, \mathbf{D}_v^T]$. The weight at a pixel i is $w_i = 1/(\sqrt{(\mathbf{D}_h \mathbf{x})_i^2 + (\mathbf{D}_v \mathbf{x})_i^2} + \epsilon)$. The weight matrix $\mathbf{W}$ contains the respective weights along its diagonal.

It has been proved [38,39,59] that the reweighted ℓ_2 formulation of TV is a majorizer [32] on the function $TV(\mathbf{x})$. Therefore, instead of minimizing $L(\mathbf{x})$, $\mathbf{x}$ is estimated by minimizing the majorizer or surrogate function which upper bounds $L(\mathbf{x})$,

$$Q(\mathbf{x}|\mathbf{x}^{(t)}) = \frac{1}{2} \sum_i \gamma_i (x_i - x_i^{ML})^2 + \frac{\lambda}{2} \mathbf{x}^T \mathbf{D}^T \mathbf{W}^{(t)} \mathbf{D} \mathbf{x}. \tag{26}$$

This means that $L(\mathbf{x}) \leq Q(\mathbf{x}|\mathbf{x}^{(t)})$, with equality when $\mathbf{x} = \mathbf{x}^{(t)}$. The monotonicity property means that

$$L(\mathbf{x}^{(t+1)}) \leq Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) \leq Q(\mathbf{x}^{(t)}|\mathbf{x}^{(t)}) = L(\mathbf{x}^{(t)}). \tag{27}$$

See [39] and [59] for the proof and a summary of the properties of majorizer functions.

In [59], the authors present conditions for convergence in terms of the second derivatives of the objective function and the majorizer, but add that these conditions are difficult to verify and further state that in practice, for convergence, the majorizer $Q(\mathbf{x}|\mathbf{x}^{(t)})$ must approximate the function $L(\mathbf{x})$ as closely as possible in the neighborhood of $\mathbf{x}^{(t)}$. In the experimental results presented in Section 4, this is verified for the case of additive and Gaussian noise in Fig. 2(a).

The MM framework refers to a class of algorithms and not a specific method, therefore analysis of convergence conditions and speed is not easy. Many MM algorithms exhibit a slow rate of convergence. Theoretical calculations for several MM algorithms are too hard to carry out, and one must rely on numerical experiments to determine the rate of convergence [59]. However, the authors of [59] defend the use of MM algorithms in spite of the uncertainties about rates of convergence because in large-scale problems, many traditional algorithms are simply infeasible.

In [54], the authors make use of the fact that the MM algorithms are a generalization of the Expectation Maximization (EM) methods [18] and retain the monotonicity and convergence properties of EM, to present a set of convergence conditions, which can be tested on the surrogate function $Q(.|.)$ and the mapping,

$$\mathbf{x}^{(t+1)} = \arg\min_\mathbf{x} Q(\mathbf{x}|\mathbf{x}^{(t)}) = F(\mathbf{x}^{(t)}), \tag{28}$$

which can be expressed as a function of the previous iteration. In our case, we have a closed form for $F(.)$, given by (22). Since the spatially adaptive operators A, B, C are functions of the coefficients $\mathbf{w}^{(t)}$, we can consider them independent of $\mathbf{x}^{(t)}$. Thus our mapping is Eqs. (21) or (22) which are used iteratively to solve (25),

$$F(\mathbf{x}^{(t)}) = A^{(t)} \mathbf{x}^{ML} + \left(B^{(t)} + C^{(t)} \right) \mathbf{x}^{(t)}. \tag{29}$$

The regularity conditions from [54] which are assumed for the convergence conditions to hold, are presented below.

1. *$L(.)$ is differentiable, with a continuous derivative.*
   From (24), it appears as though this condition is not satisfied because of the non-smooth TV term. But this problem can be got around by adding a small positive offset δ while calculating the TV function $TV(\mathbf{x}) = \sum_i \sqrt{(\mathbf{D_h x})_i^2 + (\mathbf{D_v x})_i^2 + \delta}$, thereby avoiding a possible singularity after differentiation w.r.t. $\mathbf{x}$ [39].
2. *$Q(\mathbf{x}|\mathbf{x}')$ is continuous in both $\mathbf{x}$ and $\mathbf{x}'$, and differentiable in $\mathbf{x}'$.*
   This is easily verified from (26).
3. *All the stationary points of the function $L(.)$ are isolated.*
   This is verified because in (24), the sum of two terms is convex and because of the quadratic term does not have any "ridge" along which the function $L(.)$ is stationary.

After suitable sign reversals to account for notation and the fact that the convergence conditions were defined for the EM algorithms, the conditions from [54] required for the convergence of a MM algorithm are as follows.

1. *Corollary to Theorem 2 [54]: Under the assumption that the mapping function $F(.)$ is continuous over S_l, the set of stationary points of the function $L(.)$, the sequence of iterates $\mathbf{x}^{(t)}$ is convergent i.f.f. $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\| \to 0$, when $t \to \infty$.*
   While it is verified from (29) that our recursive filter is continuous, it is not easy to show that the sequence of iterates is convergent. In the experimental results, we present the evolution of the difference between successive iterates in Fig. 2(b), where this can be verified.

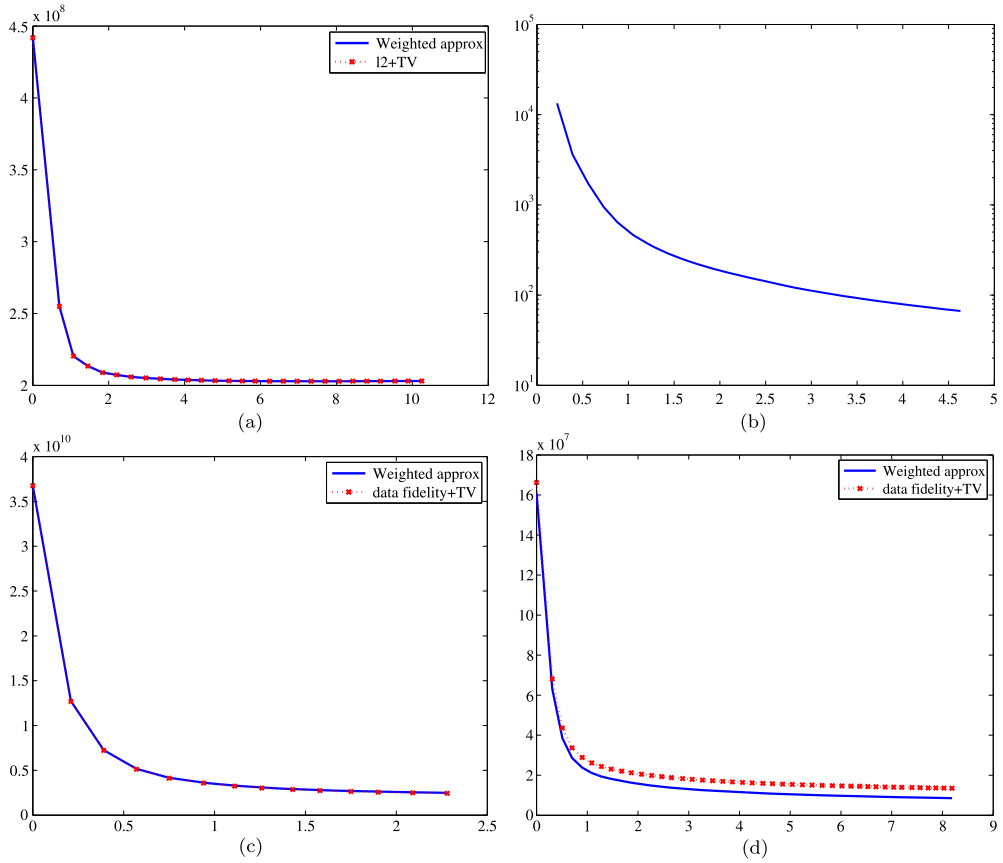

Fig. 2. Evolution of the objective function for the Lena image. (a) Evolution of the objective functions from (5) and (15) for Gaussian noise; (b) evolution of the difference between successive iterates $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|$; (c) evolution of the data fidelity term plus TV and reweighted and approximated objective function, for Rayleigh noise; (d) the respective objective functions for Poisson noise.

2. *Theorem 3 [54]: Under the assumption that for all* $\mathbf{x} \in S_l$ *there exists a unique global minimum of* $Q(.|\mathbf{x})$*, then for any starting value* $\mathbf{x}^0$ *of the sequence* $\{\mathbf{x}^{(t)}\}$, $\mathbf{x}^{(t)} \to \mathbf{x}^*$ *when* $t \to \infty$*, for some stationary point* $\mathbf{x}^* \in S_l$*. Moreover,* $F(\mathbf{x}^*) = \mathbf{x}^*$*, and if* $\mathbf{x}^{(t)} = \mathbf{x}^*$ *for all* t*, the sequence of likelihood values* $\{L(\mathbf{x}^{(t)})\}$ *is strictly decreasing to* $L(\mathbf{x}^*)$. This is the most significant result for the purpose of evaluating our framework. In the MM algorithms, it means that if the function $Q(.|\mathbf{x})$ has a global minimum, convergence is guaranteed for any starting value. For our function $Q(.|\mathbf{x})$ defined in (26), it is easy to verify that it has just one global minimum.

3. Extensions to inpainting and higher order TV

The method proposed in the previous section can be extended to the inpainting problem [37,5,53] when some pixels of the observed image **y** are missing, in a straight forward manner. This is also true for higher order TV [33], taking into account a larger neighborhood of pixels. In both cases, the only change from algorithm *ATVF* is in the adaptive filter step formulated in (18). These changes can be applied to ATVF as well as the faster version ATVFRW.

3.1. Image inpainting

Inpainting is the problem of estimating a complete image when some of the pixel values of the noisy image are unknown. This may occur in situations such as faulty imaging sensors or bit errors during transmission [29,8]. The loss of pixels is formulated by an element-wise masking operation,

$$\mathbf{y} = \mathbf{H}\mathbf{f}(\mathbf{x}), \tag{30}$$

where y is the observed image in vector notation, the observation mask **H** is an identity matrix with the diagonal elements corresponding to missing pixels set to zero, and **f** is the noisy image depending on the image **x** and the observation model.

Taking into account the observation matrix, the optimization problem for estimating **x** given **y** changes from (1) to,

$$\min_{\mathbf{x}} l(\mathbf{H}\mathbf{x}, \mathbf{y}) + \frac{\lambda}{2} TV(\mathbf{x}). \tag{31}$$

In this case we use the fact that the pixel values of **y** with the corresponding diagonal element of **H** equal to one, are the corresponding ML values, and therefore the corresponding pixels of **x** can be estimated using (18). For the missing values, the corresponding pixels of **x** are estimated from the neighboring pixels. This means that the ML term in (18) can be multiplied with the diagonal mask element h_i, leading to,

$$x_i^{MAP} = a_i h_i x_i^{ML} + b_i \sum_{\mathbf{j} \in \mathcal{N}_\mathbf{i}} x_\mathbf{j} + c_i \sum_{\mathbf{k} \in \mathcal{N}'_\mathbf{i}} w_\mathbf{k} x_\mathbf{k}, \tag{32}$$

where the terms a_i, b_i, and c_i have the same expressions as before (see (18)), except that the denominator term is now $d_i = (h_i\gamma_i + 2\lambda w_i n_b + 2\lambda \sum_{\mathbf{k} \in \mathcal{N}'_\mathbf{i}} w_\mathbf{k})$.

For the faster version ATVFRW, the equation changes to,

$$x_i^{MAP} = h_i a_i x_i^{ML} + b_i \sum_{\mathbf{k} \in (\mathcal{N}_\mathbf{i} \cup \mathcal{N}'_\mathbf{i})} x_\mathbf{k}, \tag{33}$$

where $a_i = \gamma_i / d_i$, $b_i = 4\lambda w_i / d_i$, with the denominator term $d_i = (h_i\gamma_i + 4\lambda w_i)$.

3.2. Higher order TV

This adaptive filtering framework is easily extensible to higher order total variation [33], in which second (or higher) order differences are used. For example, in the 2D case, the second order TV is,

$$\mathrm{TV}_2(\mathbf{x}) = \sum_{(m,n)} \Big((\mathbf{x}_{m,n} - \mathbf{x}_{m-1,n})^2 + (\mathbf{x}_{m,n} - \mathbf{x}_{m,n-1})^2 + (\mathbf{x}_{m,n} - \mathbf{x}_{m-2,n})^2 + (\mathbf{x}_{m,n} - \mathbf{x}_{m,n-2})^2 \Big)^{1/2}. \quad (34)$$

In this case, the only changes in the proposed filter are in the sizes of the neighborhoods $\mathcal{N}_i$ and $\mathcal{N}_i'$.

We denote the order of TV used in our, one or two, as ATVF-1 or ATVF-2, respectively.

4. Experimental results

All experiments were performed on MATLAB on a four core server Intel Xeon based server with 64 GB of RAM and running Ubuntu Linux. The code for ATVF was implemented in MATLAB with the scan over all pixels for the adaptive filter implemented in both Matlab and C++ through Mex. Existing TV based methods against which we compare our proposed method are not always available in both Matlab and C or C++. Therefore for reasons of fairness in comparison, in such situations, we present computation times for ATVF using both Matlab and C++. All tables present results which were averaged over 100 instances of the respective experiment.

We test the accuracy of the proposed method with synthetic experiments, in which we corrupt known noise-less images with noise. In this case, the criteria we use are the mean square error (MSE) and the improvement in signal to noise ratio (ISNR), for Gaussian noise, and the Median Absolute Error (MAE) [21] for non-Gaussian noise. For a known image $\mathbf{x}$, noisy observation $\mathbf{y}$ and estimate $\hat{\mathbf{x}}$, the MSE is defined as,

$$\mathrm{MSE} = \frac{1}{N_x}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2,$$

where N_x is the number of pixels. The ISNR is defined as,

$$\mathrm{ISNR} = 10\log_{10}\left(\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}\right).$$

Finally, the MAE is defined as,

$$\mathrm{MAE} = \frac{1}{N_x}\|\mathbf{x} - \hat{\mathbf{x}}\|_1,$$

and the normalized MAE is defined as,

$$\mathrm{nMAE} = \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_1}{\|\mathbf{x}\|_1}.$$

We also use the structural similarity index measure (SSIM) [57] to measure the similarity between the denoised estimate and the original noise-free image.

The values of the regularization parameter λ were obtained by hand-tuning with ATVF. For existing methods, we use the values specified by the respective authors. The stopping criterion used was until the relative difference between successive iterates $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|/\|\mathbf{x}^{(t)}\|$ fell below a certain level.

Table 2
Summary of results for comparing our method with Chambolle's algorithm and the Split Bregman Algorithm, for denoising the 512×512 Lena image, with Gaussian noise (SNR 2 dB). We see that ATVF-1 produces the best possible ISNR.

Method	Iterations	CPU time (s)		ISNR (dB)	SSIM
		C/mex	Matlab		
ATVF-1	27	4.53	241.88	**11.72**	0.822
ATVFRW-1	8	1.41	29.34	10.52	0.749
ATVF-2	8	1.46	29.48	10.31	**0.829**
ATVFRW-2	5	0.89	18.70	10.26	0.810
Chambolle	17		**1.76**	10.40	0.756
SBA	–	1.80		11.45	0.818

Table 3
Summary of results for comparing our method with Chambolle's algorithm and the Split Bregman Algorithm, for denoising the 256×256 house image, with Gaussian noise (SNR 2 dB). We see that ATVF-1 produces the best possible ISNR.

Method	Iterations	CPU time (s)		ISNR (dB)	SSIM
		C/mex	Matlab		
ATVF-1	32	2.08	68.95	**9.96**	0.638
ATVFRW-1	133	8.00	9.33	8.64	0.507
ATVF-2	6	0.47	14.8	8.82	0.571
ATVFRW-2	6	0.51	6.11	7.69	0.614
Chambolle	16		**0.51**	7.99	0.422
SBA	17	0.48		9.45	0.544

4.1. Synthetic examples

In Fig. 2 for denoising with the Lena image, we verify that the reweighted objective function closely approximates the ℓ_2+TV objective function, and that the difference between successive iterates is decreasing. We also see from Fig. 2 that for the non-Gaussian noises, the approximate reweighted model leads to an objective function that is close in value to the sum of the respective data fidelity term and the TV term, or follows the same monotonic trend.

4.1.1. Gaussian noise

For additive Gaussian noise, we compare ATVF against Chambolle's algorithm [10] and the Split Bregman Algorithm (SBA) for TV denoising [27]. Chambolle's method is implemented in Matlab but essentially uses matrix-vector multiplications, so therefore would not be significantly faster if implemented in C or C++. SBA is a very fast algorithm, being a part of the Bregman iterations or Alternating Direction Method of Multipliers (ADMM) [22,61] family of algorithms, and is available in C.

For denoising, the regularization parameter λ was set to 50 for ATVF1 and 25 for ATVF2, for both the Lena and House images. For inpainting with a random sampling mask, the values used were 20 for ATVF1 and 10 for ATVF2. For inpainting with the superimposed text, these values were 10 and 5 respectively.

Table 2 summarizes the results for denoising the 512×512 Lena image with Gaussian noise, SNR = 2 dB. We see that Chambolle's method and SBA are computationally faster, as expected, since ATVF and ATVFRW have to go over every pixel in every iteration. But ATVF-1 leads to a higher ISNR and SSIM than achievable with either. It can also be noted that with second order TV, ATVF-2 and its faster version ATVFRW-2, converge faster but do not produce as good a value of ISNR as ATVF-1. The original image is shown in Fig. 3(a), and the noisy observed image is shown in Fig. 3(b). Figs. 3(c) to 3(h) present the denoised estimates. A similar trend is observed for the House image, which is illustrated in Fig. 4, with the results summarized in Table 3.

For the inpainting problem, we compare our method with two methods that can use TV regularization, the Split Augmented La-

Fig. 3. Lena image with Gaussian noise (SNR = 2 dB): (a) original, (b) noisy; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) Chambolle's method, (h) split-Bregman algorithm.

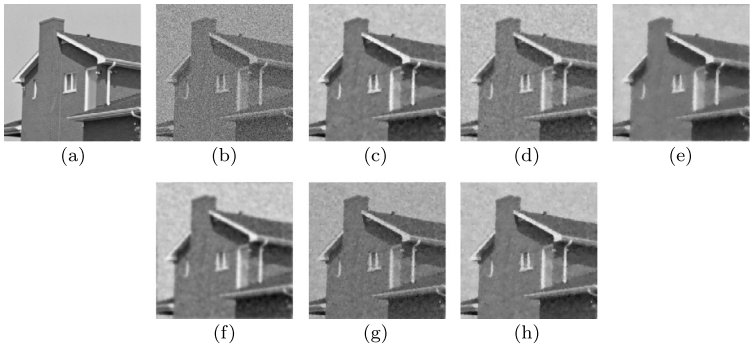

Fig. 4. House image with Gaussian noise (SNR = 2 dB): (a) original, (b) noisy; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) Chambolle's method, (h) split-Bregman algorithm.

Fig. 5. Inpainting with the Lena image, Gaussian noise: (a) original, (b) original with additive Gaussian noise; (c) noisy image with 50% of the pixels missing; estimates using: (d) ATVF-1, (e) ATVFRW-1, (f) ATVF-2, (g) ATVFRW-2, (h) SALSA, (i) FISTA.

Table 4
Summary of comparison of our method with other inpainting methods, for inpainting the 512 × 512 Lena image, with Gaussian noise (SNR 10 dB) and 50% of the pixels missing. We see that ATVFRW is faster than the existing methods and produces a higher ISNR, even though the best ISNR is produced by ATVF.

Method	Iterations	CPU time (s)	ISNR (dB)	SSIM
ATVF-1	52	201.97	**21.00**	**0.879**
ATVFRW-1	39	**68.05**	20.22	0.871
ATVF-2	39	150.25	18.90	0.862
ATVFRW-2	36	**68.2**	18.64	0.861
SALSA	323	101.90	18.22	0.787
FISTA	121	103.27	18.20	0.787

Table 5
Summary of comparison of our method with other inpainting methods, for inpainting the House image, with Gaussian noise (SNR 10 dB) and 50% of the pixels missing. We see that ATVFRW is faster than the existing methods and produces a higher ISNR, even though the best ISNR is produced by ATVF.

Method	Iterations	CPU time (s)	ISNR (dB)	SSIM
ATVF-1	52	204.51	**21.00**	0.879
ATVFRW-1	39	**69.74**	20.22	0.871
ATVF-2	39	150.17	18.90	0.862
ATVFRW-2	36	**68.34**	18.64	0.861
SALSA	320	94.56	18.22	0.787
FISTA	120	102.12	18.20	0.787

grangian and Shrinkage Algorithm (SALSA) [1] and the Fast Iterative Shrinkage and Thresholding Algorithm (FISTA) [4]. In this case, since both methods have Matlab implementations available, we present computation times only for the Matlab implementation of our method. Fig. 5(a) shows the Lena image with Gaussian noise, SNR = 10 dB, and Fig. 5(c) shows the observed image with 50% of its pixels randomly discarded and set to zero. Figs. 5(d) to 5(i) show the estimates obtained from Fig. 5(c). The computation times, ISNR, and SSIM values are summarized in Table 4. We see that both ATVF and ATVFRW produce better ISNRs than SALSA or

Fig. 6. Inpainting with the Lena image, Gaussian noise: (a) original, (b) noisy image with 80% of the pixels missing; estimates using: (d) ATVF-1, (e) ATVF-2, (f) SALSA, (g) FISTA.

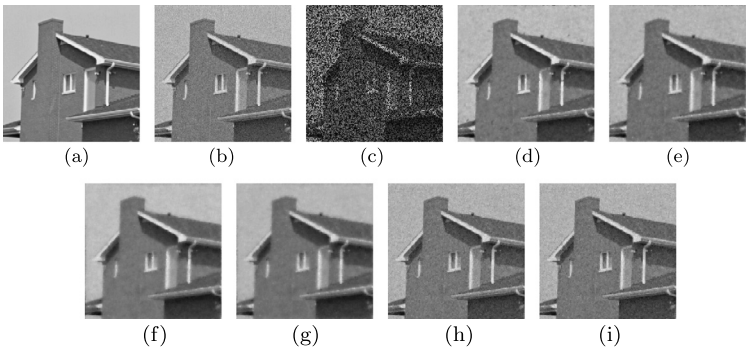

Fig. 7. Inpainting with the House image, Gaussian noise: (a) original, (b) original with additive Gaussian noise; (c) noisy image with 50% of the pixels missing; estimates using: (d) ATVF-1, (e) ATVFRW-1, (f) ATVF-2, (g) ATVFRW-2, (h) SALSA, (i) FISTA.

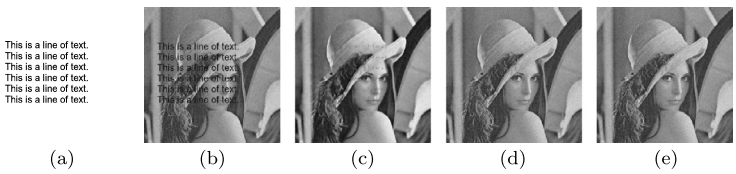

Fig. 8. Inpainting with the Lena image, Gaussian noise with superimposed text: (a) sampling mask, (b) observed image; estimates using: (c) ATVF-1, (d) SALSA, (e) FISTA.

Table 6
Summary of comparison of our method with other inpainting methods, for inpainting the Lena image, with Gaussian noise (SNR 5 dB) and 80% of the pixels missing. We see that ATVF2 produces the best ISNR when the fraction of missing pixels is high.

Method	Iterations	CPU time (s)	ISNR (dB)	SSIM
ATVF-1	95	167.6	15.81	0.648
ATVF-2	70	113.17	**17.60**	0.731
SALSA	342	269.30	15.59	0.576
FISTA	220	189.61	16.94	0.576

FISTA, and ATVFRW is computationally the fastest. The same is observed for the House image in Fig. 7 and Table 5.

We present another example with a high proportion of the pixels missing. In Fig. 6, we illustrate inpainting with the Lena image with 80% of its pixels missing, and present the estimates obtained with ATVF1, ATVF2, SALSA, and FISTA. We can see from the results in Table 6, that in this case with a high loss of pixels, ATVF2 produces a higher ISNR than ATVF1. Therefore, when we have high fractions of pixels missing, using a higher order TV is advantageous.

We present an example with inpainting when the sampling mask is not random, but text superimposed on the image. Fig. 8 shows the sampling mask, observed image, and estimates obtained using ATVF-1, SALSA, and FISTA. We can see from the results summarized in Table 7 that our method is slower than SALSA and FISTA but produces a higher ISNR.

Table 7
Summary of comparison of our method with other inpainting methods, for inpainting the Lena image with Gaussian noise and text superimposed. We see that ATVF is slower than the existing methods but produces a better ISNR.

Method	Iterations	CPU time (s)	ISNR (dB)	SSIM
ATVF-1	66	295.43	**12.19**	**0.902**
SALSA	180	53.41	7.43	0.784
FISTA	129	110.13	7.67	0.783

4.1.2. Poisson noise

For Poisson noise, our comparison is with Poisson Image Denoising using Augmented Lagrangian (PIDAL) [25] and Log TV [43] for the denoising problem. For the inpainting problem, our comparison is with PIDAL, which, like SBA and SALSA, is an augmented Lagrangian/ADMM based method. Since PIDAL was implemented in Matlab, and Log TV in C, once again, we present computation times using both our implementations in Table 8.

Expectedly, PIDAL was found to be the fastest method among the Matlab implementations, which led to an ISNR above 8 dB. However, ATVF-1 produced the highest ISNR, and its C++ implementation was computationally faster than LogTV. The noisy image and estimates obtained using the various methods are presented in

Table 8
Summary of results for denoising the 512 × 512 Lena image, with Poisson noise. ATVF1 produces the best ISNR, although PIDAL, an ADMM method, is faster.

Method	Iterations	CPU time (s)		MAE	Normalized MAE	ISNR (dB)	SSIM
		C/mex	Matlab				
ATVF-1	6	1.19	54.58	1.418	1.748e–07	**9.07**	**0.977**
ATVFRW-1	5	1.07	17.43	1.825	2.245e–07	7.15	0.967
ATVF-2	4	0.92	38.41	2.21	2.723e–07	5.6	0.968
ATVFRW-2	4	0.77	14.24	1.85	2.28e–07	6.68	0.971
PIDAL	16	–	**17.43**	**1.41**	**1.73e–07**	8.6	0.975
Log TV	9	1.06	–	1.475	1.815e–07	8.79	0.974

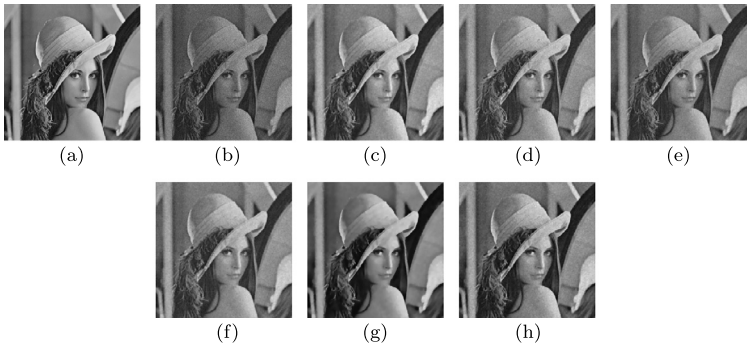

Fig. 9. Lena image with Poisson noise: (a) original, (b) noisy; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) PIDAL, (h) Log TV.

Table 9
Summary of results for denoising the House image, with Poisson noise.

Method	Iterations	CPU time (s)		MAE	Normalized MAE	ISNR (dB)	SSIM
		C/mex	Matlab				
ATVF-1	6	0.42	13.72	1.511	6.68e–07	8.96	0.966
ATVFRW-1	5	0.39	4.45	1.939	8.58e–07	7.07	0.941
ATVF-2	4	0.36	9.19	2.312	1.023e–06	5.65	0.906
ATVFRW-2	5	0.44	3.63	2.028	8.97e–07	6.25	0.940
PIDAL	16	–	4.17	1.37	**6.058e–07**	**9.16**	**0.968**
Log TV	8	0.25	–	1.552	6.86e–07	8.88	0.962

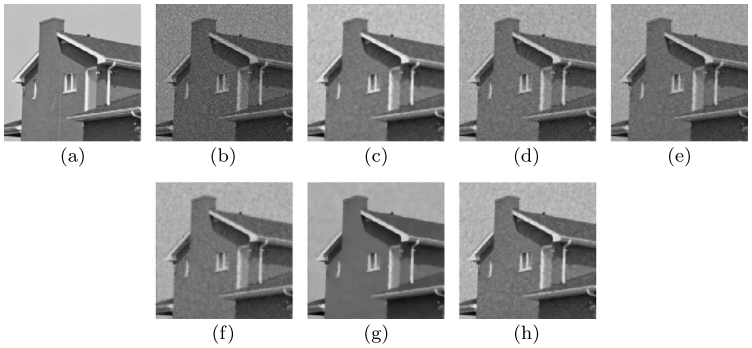

Fig. 10. House image with Poisson noise: (a) original, (b) noisy; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) PIDAL, (h) Log TV.

Fig. 9. For the House image, ATVF1 produced an ISNR close to that obtained by PIDAL, as summarized in Table 9. The observed image and estimates are presented in Fig. 10.

Table 10 and Fig. 11 present the results for inpainting with Poisson noise. Even though ATVFRW-1 reached the maximum number of iterations, it produced the most accurate result. For the House image, the tendencies are similar as summarized in Table 11, but the best value of the ISNR is produced by ATVF1.

For denoising, the regularization parameter λ was set to 1 for ATVF1 and 0.025 for ATVF2, for both the Lena and House images. For inpainting with a random sampling mask, the values used were 0.05 for ATVF1 and 0.025 for ATVF2 (Figs. 11 and 12).

4.1.3. Rayleigh speckle noise

For the Rayleigh multiplicative noise case, our comparison is with Rayleigh Log TV (RLTV) [50,49] and with the ADMM based reconstruction method [2] for Rayleigh multiplicative noise. Both these methods were implemented in Matlab.

From Tables 12 and 14, we see that ATVFRW-1 produces the lowest MAE value for both denoising and inpainting. The second order TV methods are faster for the denoising problem, but did not produce a visually reasonably good estimate for the inpainting problem and were therefore not included in the comparison. The observed and estimated images are presented in Fig. 13 for the denoising problem, and in Fig. 15 for the inpainting problem. Tables 13 and 15 present the corresponding results for inpainting with the House image.

Table 10
Summary of results for inpainting with the 512 × 512 Lena image, with Poisson noise and 50% of the pixels missing. The best ISNR was obtained using ATVF2.

Method	Iterations	CPU time (s)	MAE	Normalized MAE	ISNR (dB)	SSIM
ATVF-1	59	232.15	2.212	2.721e–07	18.03	0.957
ATVFRW-1	13	23.51	2.241	2.756e–07	17.85	0.949
ATVF-2	49	188.98	2.117	**2.601e–07**	**18.14**	**0.964**
ATVFRW-2	73	268.38	3.425	4.213e–07	13.97	0.928
PIDAL	12	**3.07**	3.319	4.83e–07	15.24	0.948

Fig. 11. Inpainting with the Lena image with Poisson noise: (a) original, (b) noisy image with 50% of the pixels missing; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) PIDAL.

Table 11
Summary of results for inpainting with the House image, with Poisson noise and 50% of the pixels missing. The best ISNR was obtained using ATVF1.

Method	Iterations	CPU time (s)	MAE	Normalized MAE	ISNR (dB)	SSIM
ATVF-1	60	119.57	2.319	**1.026e–06**	**18.26**	0.922
ATVFRW-1	13	12.55	2.202	1.063e–06	17.94	0.917
ATVF-2	50	96.02	2.344	1.037e–06	17.74	**0.935**
ATVFRW-2	72	70.03	3.91	1.733e–06	13.23	0.912
PIDAL	12	**0.03**	3.58	1.59e–06	15.38	0.928

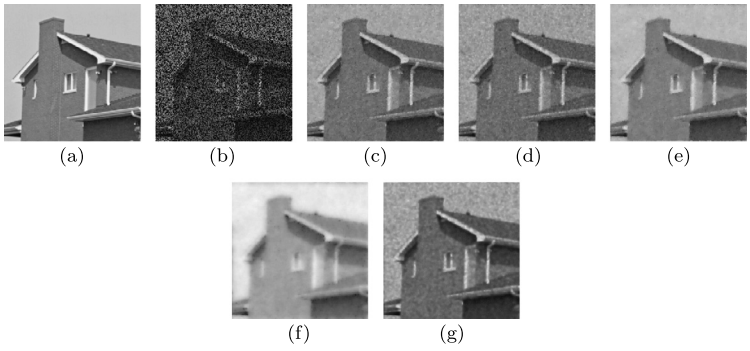

Fig. 12. Inpainting the House image with Poisson Noise: (a) original, (b) noisy image with 50% of the pixels missing; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) PIDAL.

Table 12
Summary of results for denoising the 512 × 512 Lena image, with Rayleigh noise. The best estimate in terms of ISNR is produced by ATVFRW1.

Method	Iterations	CPU time (s)	Normalized MAE	SSIM
ATVF-1	10.75	91.70	1.9e–06	0.991
ATVFRW-1	68	12.43	**1.2e–06**	**0.996**
ATVF-2	8	74.31	1.5e–06	0.994
ATVFRW-2	18	**3.43**	1.3e–06	0.995
RLTV	11	277.20	2.4e–06	0.984

Table 13
Summary of results for denoising the House image, with Rayleigh noise. The best estimate in terms of ISNR is produced by ATVFRW1.

Method	Iterations	CPU time (s)	Normalized MAE	SSIM
ATVF-1	11	24.64	7.2e–06	0.989
ATVFRW-1	71	4.04	**3.3e–06**	**0.997**
ATVF-2	10	22.84	4.6e–06	0.995
ATVFRW-2	21	1.57	3.5e–06	0.997
RLTV	35	220.25	9.2e–06	0.982

Tables 12 and 14 show that the trend is similar for the house image, with the respective observed images and estimates presented in Figs. 14 and 16.

For denoising, the regularization parameter λ was set to 0.1 for ATVF1 and 0.025 for ATVF2, for both the Lena and House images. Note that the image is normalized in this case. For inpainting with

Fig. 13. Lena image with Rayleigh multiplicative noise: (a) original, (b) noisy; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) RLTV.

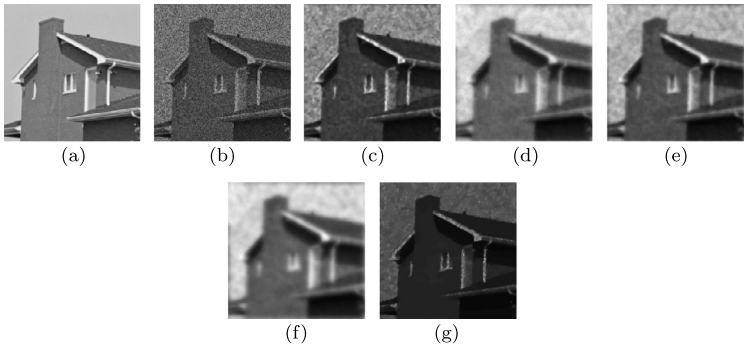

Fig. 14. House image with Rayleigh multiplicative noise: (a) original, (b) noisy; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ATVF-2, (f) ATVFRW-2, (g) RLTV.

Fig. 15. Inpainting with the Lena image with Rayleigh noise: (a) original, (b) noisy image with 50% of the pixels missing; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ADMM.

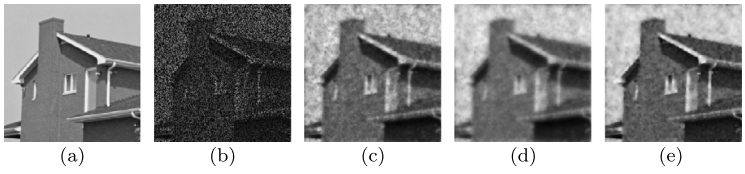

Fig. 16. Inpainting with the House image with Rayleigh noise: (a) original, (b) noisy image with 50% of the pixels missing; estimates using: (c) ATVF-1, (d) ATVFRW-1, (e) ADMM.

Table 14
Summary of results for inpainting with the 512×512 Lena image, with Rayleigh noise and 50% of the pixels missing. The best estimate in terms of ISNR is produced by ATVFRW1.

Method	Iterations	CPU time (s)	MAE	Normalized MAE	SSIM
ATVF-1	37	290.64	0.16	1.21e–06	0.996
ATVFRW-1	80	286.66	0.1345	**9.8e–07**	0.997
ADMM	5	1.96	0.5063	3.815e–06	0.958

Table 15
Summary of results for inpainting with the House image, with Rayleigh noise and 50% of the pixels missing. The best estimate in terms of ISNR is produced by ATVFRW1.

Method	Iterations	CPU time (s)	MAE	Normalized MAE	SSIM
ATVF-1	37	72.83	0.132	3.37e–06	0.997
ATVFRW-1	81	73.96	0.1086	**2.871e–06**	**0.998**
ADMM	41	3.72	0.1235	3.465e–06	0.997

a random sampling mask, the value used was 2 for both ATVF1 and ATVF2 (Table 15).

4.2. Real data

When we wish to denoise acquired biomedical images, we do not have access to the ground truth. Therefore, our comparison with other algorithms in this situation is on the basis of computation time only after optimizing the respective parameters of each method to produce a visually good image.

For multiplicative Rayleigh noise, we have a transversal ultrasound (US) image of the carotid artery, acquired using a Philips Medical Systems ATL HDI 5000 Ultrasound machine. The extracted radio frequency (RF) envelope image is shown in Fig. 17(a), after

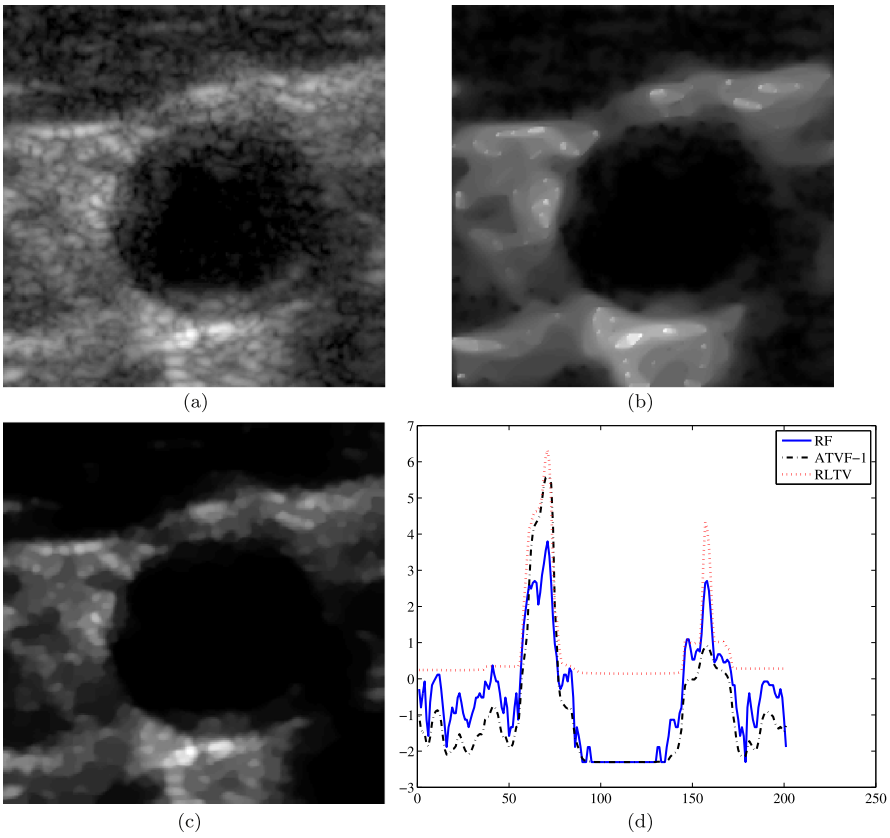

Fig. 17. Transversal US image of the carotid artery (after contrast enhancement for display): (a) RF envelope image with speckle, estimates using: (b) ATVF-1, (c) RLTV. (d) diagonal profiles of the noisy and estimated images.

Table 16
Denoising real biomedical images – summary.

Carotid artery – transversal US image size: 201 × 201			Immunofluorescence image size: 170 × 232 × 3			
Method	Iterations	CPU time (s)	Method	Iterations	CPU time (s)	
					C/mex	Matlab
ATVF-1	105	75.78	ATVF-1	5	0.37	**6.49**
RLTV	47	**31.30**	PIDAL	21	–	8.09
			Log TV	9	**0.23**	–

contrast enhancement for display. The estimates using ATVF-1, and RLTV are shown in Figs. 17(b), and 17(c), respectively. Fig. 17(d) shows the diagonal profiles of the speckled and estimated images. From Table 16, we see that the Matlab version of ATVF-1 was not the computationally fastest method, however from the diagonal profiles we see that for lower values (the figure is shown in the logarithmic scale), it preserves values better than RLTV.

For the Poisson model, we perform the denoising of a fluorescence microscopy image of epithelial cells of the stomach with tagged E-Cadherin [23]. The image size is 240 × 320. The estimates using ATVF-1, PIDAL, and Log TV are shown in, respectively, Figs. 18(b), 18(c), and 18(d). Figs. 18(e) and 18(f) show the diagonal profiles of the noisy and estimated images, for the green and blue components, respectively. From the computation times presented in Table 16, we see that ATVF-1 is slower than Log TV for the C/C++ versions, but its Matlab version is faster than PIDAL. Note that the number of iterations and the CPU times were the sums of those for the green and blue components of the color image. It is obvious that the red component is negligible.

5. Conclusions

We have proposed a Bayesian MAP estimation method to solve the TV denoising problem in a common framework for different statistical models. The proposed method, ATVF, involves recursive spatially adaptive filtering and can be easily extended to inpainting and higher order TV. This method falls within the class of MM algorithms for which convergence conditions can be verified. Experimental results show that ATVF is quite competitive with existing denoising and inpainting methods, and is therefore applicable as a solver for the denoising problems with non-Gaussian observation models.

Directions of current and future work include:

- Continuation on the parameter ϵ, through the use of appropriate surrogate functions, leading to another step at each iteration,

$$\epsilon^{(t+1)} = \arg\min_{\epsilon} l(\mathbf{x}^{(t)}, \mathbf{W}^{(t)}, \epsilon).$$

  Another idea worth investigating would be to treat ϵ as a regularization parameter on the weights, and study the relation between the proposed method which has an IRLS-MM flavor, with the ADMM framework.
- Speeding up the algorithm as can be done for MM algorithms using a quasi-Newton method [59,32].
- Optimizing the scan pattern through the pixels to reduce computation time, and facilitate parallelization and implementation on GPUs.

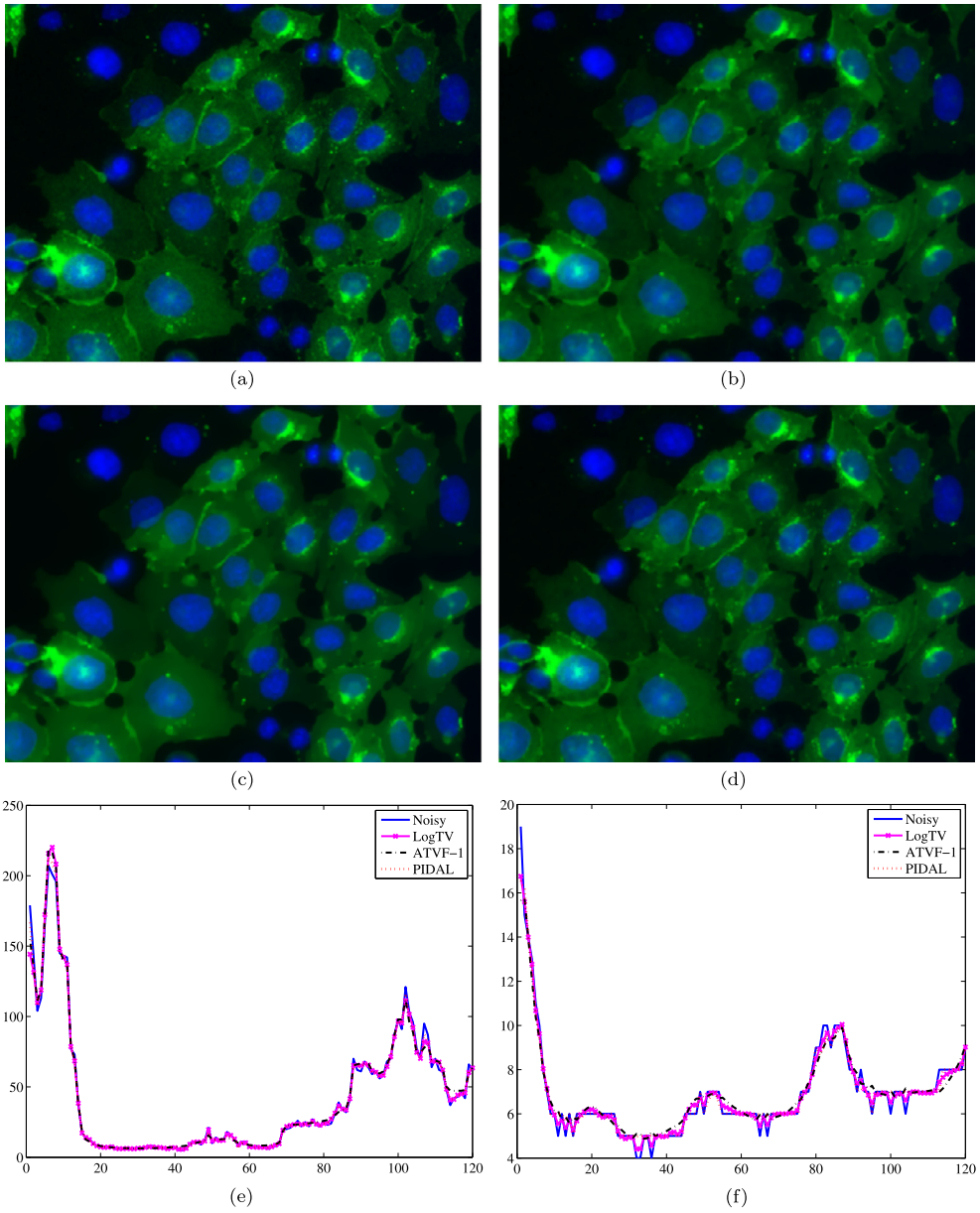

Fig. 18. Immunofluorescence image: (a) acquired image with Poisson noise, estimates using: (b) ATVF-1, (c) PIDAL, (d) Log TV. (e) and (f): diagonal profiles of the noisy and estimated images for: (e) green component, (f) blue component. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Acknowledgments

This work was supported by *Fundação para a Ciência e Tecnologia* (FCT), Portuguese Ministry of Science and Higher Education, through a Post-doctoral fellowship (contract No. SFRH/BPD/79011/2011) and FCT project (FCT [UID/EEA/5009/2013]).

The carotid artery transversal US images were acquired at the Instituto Cardio-Vascular de Lisboa (ICVL). The immunofluorescence images were acquired at the Institute of Molecular Pathology and Immunology at the University of Porto (IPATIMUP).

References

[1] M. Afonso, J. Bioucas-Dias, M. Figueiredo, Fast image recovery using variable splitting and constrained optimization, IEEE Trans. Image Process. 19 (9) (2010) 2345–2356.

[2] M. Afonso, J. Sanches, A total variation based reconstruction algorithm for 3D ultrasound, in: Pattern Recognition and Image Analysis, in: Lecture Notes in Computer Science, vol. 7887, Springer, Berlin/Heidelberg, 2013, pp. 149–156.

[3] F. Agostinelli, M.R. Anderson, H. Lee, Adaptive multi-column deep neural networks with application to robust image denoising, in: Advances in Neural Information Processing Systems, 2013, pp. 1493–1501.

[4] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.

[5] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[6] J. Bioucas-Dias, M. Figueiredo, A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration, IEEE Trans. Image Process. 16 (12) (2007) 2992–3004.

[7] J. Bioucas-Dias, M. Figueiredo, Multiplicative noise removal using variable splitting and constrained optimization, IEEE Trans. Image Process. 19 (7) (2010) 1720–1730.

[8] A.C. Bovik, Handbook of Image and Video Processing, Academic Press, 2010.

[9] E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate information, Commun. Pure Appl. Math. 59 (8) (2005) 1207–1233.

[10] A. Chambolle, An algorithm for total variation minimization and applications, J. Math. Imaging Vis. 20 (1) (2004) 89–97.

[11] T. Chan, S. Esedoglu, F. Park, A. Yip, Recent developments in total variation image restoration, in: Handbook of Mathematical Models in Computer Vision, 2005, pp. 17–30.

[12] R. Chartrand, V. Staneva, Total variation regularisation of images corrupted by non-Gaussian noise using a quasi-newton method, IET Image Process. 2 (6) (2008) 295–303.
[13] R. Chartrand, Nonconvex compressive sensing and reconstruction of gradient-sparse images: random vs. tomographic Fourier sampling, in: 15th IEEE International Conference on Image Processing, ICIP 2008, IEEE, 2008, pp. 2624–2627.
[14] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008, IEEE, 2008, pp. 3869–3872.
[15] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[16] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math. 57 (11) (2004) 1413–1457.
[17] I. Daubechies, R. DeVore, M. Fornasier, C.S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, Commun. Pure Appl. Math. 63 (1) (2010) 1–38.
[18] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. B (1977) 1–38.
[19] D. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[20] D.L. Donoho, Y. Tsaig, Fast solution of ℓ_1-norm minimization problems when the solution may be sparse, IEEE Trans. Inf. Theory 54 (11) (2008) 4789.
[21] S. Durand, J. Fadili, M. Nikolova, Multiplicative noise removal using ℓ_1 fidelity on frame coefficients, J. Math. Imaging Vis. 36 (3) (2010) 201–226.
[22] J. Eckstein, D. Bertsekas, On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators, Math. Program. 55 (3) (1992) 293–318.
[23] S. Esménio, J. Figueiredo, R. Seruca, J. Sanches, E-cadherin radial distribution characterization for mutation detection purposes, in: Pattern Recognition and Image Analysis, in: Lecture Notes in Computer Science, vol. 7887, Springer, Berlin/Heidelberg, 2013, pp. 173–180.
[24] M. Figueiredo, J. Bioucas-Dias, R. Nowak, Majorization–minimization algorithms for wavelet-based image restoration, IEEE Trans. Image Process. 16 (12) (2007) 2980–2991.
[25] M.A. Figueiredo, J.M. Bioucas-Dias, Restoration of poissonian images using alternating direction optimization, IEEE Trans. Image Process. 19 (12) (2010) 3133–3145.
[26] J. Gao, Q. Chen, E. Blasch, Image denoising in the presence of non-Gaussian, power-law noise, in: IEEE National Aerospace and Electronics Conference, NAECON 2012, IEEE, 2012, pp. 103–108.
[27] T. Goldstein, S. Osher, The split Bregman method for ℓ_1 regularized problems, SIAM J. Imaging Sci. 2 (2) (2009) 323–343.
[28] G.H. Golub, P.C. Hansen, D.P. O'Leary, Tikhonov regularization and total least squares, SIAM J. Matrix Anal. Appl. 21 (1) (1999) 185–194.
[29] R.C. Gonzalez, R.E. Woods, S.L. Eddins, Digital Image Processing Using MATLAB, vol. 2, Gatesmark Publishing, Tennessee, 2009.
[30] I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using focus: a re-weighted minimum norm algorithm, IEEE Trans. Signal Process. 45 (3) (1997) 600–616.
[31] R. Gribonval, M. Nielsen, Highly sparse representations from dictionaries are unique and independent of the sparseness measure, Appl. Comput. Harmon. Anal. 22 (3) (2007) 335–355.
[32] D. Hunter, K. Lange, A tutorial on MM algorithms, Am. Stat. 58 (1) (2004) 30–37.
[33] F. Knoll, K. Bredies, T. Pock, R. Stollberger, Second order total generalized variation (TGV) for MRI, Magn. Reson. Med. 65 (2) (2011) 480–491.
[34] C.L. Lawson, Contributions to the theory of linear least maximum approximation, Ph.D. dissertation, University of California, Los Angeles-Mathematics, 1961.
[35] Y. Li, A globally convergent method for l_p problems, SIAM J. Optim. 3 (3) (1993) 609–629.
[36] J.V. Manjón, P. Coupé, L. Concha, A. Buades, D.L. Collins, M. Robles, Diffusion weighted image denoising using overcomplete local PCA, PLoS ONE 8 (9) (2013) e73021.
[37] S. Masnou, J.-M. Morel, Level lines based disocclusion, in: Proceedings of International Conference on Image Processing, ICIP 98, IEEE, 1998, pp. 259–263.
[38] N. Mourad, J.P. Reilly, lp minimization for sparse vector reconstruction, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009, IEEE, 2009, pp. 3345–3348.
[39] J.P. Oliveira, J.M. Bioucas-Dias, M.A. Figueiredo, Adaptive total variation image deblurring: a majorization–minimization approach, Signal Process. 89 (9) (2009) 1683–1693.
[40] M.R. Osborne, Finite Algorithms in Optimization and Data Analysis, John Wiley & Sons, Inc., 1985.
[41] I. Ram, M. Elad, I. Cohen, Image denoising using NL-means via smooth patch ordering, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, May 2013, pp. 1350–1354.
[42] B.D. Rao, K. Kreutz-Delgado, An affine scaling methodology for best basis selection, IEEE Trans. Signal Process. 47 (1) (1999) 187–200.
[43] I.C. Rodrigues, J.M.R. Sanches, Convex total variation denoising of Poisson fluorescence confocal images with anisotropic filtering, IEEE Trans. Image Process. 20 (1) (2011) 146–160.
[44] P. Rodríguez, Total variation regularization algorithms for images corrupted with different noise models: a review, Int. J. Electr. Comput Syst. Eng. 2013 (2013), Article 21702 .
[45] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.
[46] J. Salmon, Z. Harmany, C.-A. Deledalle, R. Willett, Poisson noise reduction with non-local PCA, J. Math. Imaging Vis. 48 (2) (2014) 279–294.
[47] J.M. Sanches, J.S. Marques, A fast map algorithm for 3D ultrasound, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, 2001, pp. 63–74.
[48] J.M. Sanches, J.S. Marques, A map estimation algorithm using IIR recursive filters, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, 2003, pp. 436–449.
[49] J. Seabra, J. Sanches, Modeling log-compressed ultrasound images for radio frequency signal recovery, in: Engineering in Medicine and Biology Society, 2008.
[50] J. Seabra, J. Sanches, Modeling log-compressed ultrasound images for radio frequency signal recovery, in: Conf. Proc. IEEE Eng. Med. Biol. Soc., 2008, pp. 426–429.
[51] J. Seabra, J. Sanches, L. Pedro, et al., Carotid plaque 3D compound imaging and echo-morphology analysis: a Bayesian approach, in: Engineering in Medicine and Biology Society, 2007, pp. 763–766.
[52] J. Seabra, J. Xavier, J. Sanches, Convex ultrasound image reconstruction with log-Euclidean priors, in: Engineering in Medicine and Biology Society, EMBS'2008, IEEE, 2008, pp. 435–438.
[53] J. Shen, T.F. Chan, Mathematical models for local nontexture inpaintings, SIAM J. Appl. Math. 62 (3) (2002) 1019–1043.
[54] F. Vaida, Parameter convergence for em and mm algorithms, Stat. Sin. 15 (3) (2005) 831.
[55] C.R. Vogel, M.E. Oman, Iterative methods for total variation denoising, SIAM J. Sci. Comput. 17 (1) (1996) 227–238.
[56] Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, SIAM J. Imaging Sci. 1 (3) (2008) 248–272.
[57] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[58] L. Wasserman, All of Statistics: A Concise Course in Statistical Inference, Springer, 2004.
[59] T.T. Wu, K. Lange, The mm alternative to em, Stat. Sci. 25 (4) (2010) 492–505.
[60] J. Xie, L. Xu, E. Chen, Image denoising and inpainting with deep neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 341–349.
[61] W. Yin, S. Osher, D. Goldfarb, J. Darbon, Bregman iterative algorithms for ℓ_1 minimization with applications to compressed sensing, SIAM J. Imaging Sci. 1 (1) (2008) 143–168.

Manya V. Afonso finished his Ph.D. at Instituto Superior Tecnico in 2011, while being a researcher at the Instituto de Telecomunicacoes, Lisboa. He is currently a Post-doc at the Instituto de Sistemas e Robotica in Instituto Superior Tecnico.

He previously received the Bachelor of Engineering degree in Electronics and Telecommunication Engineering from Goa University, India in 2003 and Master of Technology in Communication Engineering from the Indian Institute of Technology Delhi in 2005.

Manya Afonso's research interests include image processing and analysis, inverse problems, optimization, machine learning, computer vision, and video surveillance.

João M.R. Sanches teaches at the Department of Bioengineering (DBE) from the Instituto Superior TAccnico and before he was at the Department of Electrical and Computer Engineering where has taught in the area of signal processing, systems and control. J.S. has been actively involved in the course of Biomedical Engineering advising master and Ph.D. thesis on the area of Bioengineering, specially, on Biomedical engineering.

J.S. is researcher at the Institute for Systems and Robotics (ISR) in the Signal and Image Processing group (SIPg). His work has been focused in Biomedical Engineering (BME), namely, in biological and medical image processing and statistical signal processing of physiological and behavioral data. Morphological and textural characterization of tissues from Ultrasound (US) images are central issues but he is also working with fluorescence images of microscopy for biological quantification purposes. J.S. is also involved in the development of signal processing algorithms for polysomnography data and smartphones applications for long term monitoring for sleep disorders diagnosis purposes. In this scope, Heart Rate Variability analysis is today one of his main interests.

Almost all of his research work is in collaboration with medical and biological institutions, namely, Medical School of the University of Lisbon (FMUL), the Institute of Molecular Pathology and Immunology of the University of Porto (IPATIMUP) and the Electroencephalography and Clinical Neurophysiology Center (Centro de Electroencefalografia Neurofisiologia Clínica–CENC). Several publications and patents were already produced in the scope of this collaborative work.

J.S. is senior member of the IEEE Engineering in Medicine and Biology Society since 2011 and Member of the Bio Imaging and Signal Processing Technical Committee (BISP-TC) of the IEEE Signal Processing Society.