Journal of
*Imaging*

*Article*

# Directional TGV-Based Image Restoration under Poisson Noise

**Daniela di Serafino [1], Germana Landi [2,*] and Marco Viola [3]**

[1] Department of Mathematics and Applications "R. Caccioppoli", University of Naples Federico II, 80126 Naples, Italy; daniela.diserafino@unina.it
[2] Department of Mathematics, University of Bologna, 40126 Bologna, Italy
[3] Department of Mathematics and Physics, University of Campania "L. Vanvitelli", 81100 Caserta, Italy; marco.viola@unicampania.it
[*] Correspondence: germana.landi@unibo.it

**Abstract:** We are interested in the restoration of noisy and blurry images where the texture mainly follows a single direction (i.e., directional images). Problems of this type arise, for example, in microscopy or computed tomography for carbon or glass fibres. In order to deal with these problems, the Directional Total Generalized Variation (DTGV) was developed by Kongskov et al. in 2017 and 2019, in the case of impulse and Gaussian noise. In this article we focus on images corrupted by Poisson noise, extending the DTGV regularization to image restoration models where the data fitting term is the generalized Kullback–Leibler divergence. We also propose a technique for the identification of the main texture direction, which improves upon the techniques used in the aforementioned work about DTGV. We solve the problem by an ADMM algorithm with proven convergence and subproblems that can be solved exactly at a low computational cost. Numerical results on both phantom and real images demonstrate the effectiveness of our approach.

**Keywords:** directional image restoration; Poisson noise; DTGV regularization; ADMM method

**Citation:** di Serafino, D.; Landi, G.; Viola, M. Directional TGV-Based Image Restoration under Poisson Noise. *J. Imaging* **2021**, *7*, 99. https://doi.org/10.3390/jimaging7060099

Academic Editor: Nicolas Papadakis

Received: 30 April 2021
Accepted: 11 June 2021
Published: 16 June 2021

## 1. Introduction

Poisson noise appears in processes where digital images are obtained by the count of particles (generally photons). This is the case of X-ray computed tomography, positron emission tomography, confocal and fluorescence microscopy and optical/infrared astronomical imaging, to name just a few applications (see, e.g., [1] and the references therein). In this case, the object to be restored can be represented as a vector $\mathbf{u} \in \mathbb{R}^n$ and the data can be assumed to be a vector $\mathbf{b} \in \mathbb{N}_0^n$, whose entries $b_j$ are sampled from $n$ independent Poisson random variables $B_j$ with probability

$$P(B_j = b_j) = \frac{e^{-(A\mathbf{u}+\gamma)_j}(A\mathbf{u}+\gamma)_j^{b_j}}{b_j!}.$$

The matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ models the observation mechanism of the imaging system and the following standard assumptions are made:

$$a_{ij} \geq 0 \text{ for all } i, j, \qquad \sum_{i=1}^{n} a_{ij} = 1 \text{ for all } j. \tag{1}$$

The vector $\gamma \in \mathbb{R}^n$, with $\gamma > 0$, models the background radiation detected by the sensors.

By applying a maximum-likelihood approach [1,2], we can estimate $\mathbf{u}$ by minimizing the Kullback–Leibler (KL) divergence of $A\mathbf{u} + \gamma$ from $\mathbf{b}$:

$$D_{KL}(A\mathbf{u}+\gamma, \mathbf{b}) = \sum_{i=1}^{n}\left(b_i \ln \frac{b_i}{[A\mathbf{u}+\gamma]_i} + [A\mathbf{u}+\gamma]_i - b_i\right), \tag{2}$$

where we set

$$b_i \ln \frac{b_i}{[A\mathbf{u}+\gamma]_i} = 0 \quad \text{if } b_i = 0.$$

Regularization is usually introduced in (2) to deal with the ill-conditioning of this problem. The Total Variation (TV) regularization [3] has been widely used in this context, because it preserves edges and is able to smooth flat areas of the image. However, since it may produce staircase artifacts, other TV-based regularizers have been proposed. For example, the Total Generalized Variation (TGV) has been proposed and applied in [4–7] to overcome the staircasing effect while keeping the ability of identifying edges. On the other hand, to improve the quality of restoration for directional images, the Directional TV (DTV) regularization has been considered in [8], in the discrete setting. In [9,10], a regularizer combining DTV and TGV, named Directional TGV (DTGV), has been successfully applied to directional images affected by impulse and Gaussian noise.

Given an image $\mathbf{u} \in \mathbb{R}^n$, the discrete second-order Directional TGV of $\mathbf{u}$ is defined as

$$\mathrm{DTGV}^2(\mathbf{u}) = \min_{\mathbf{w}\in\mathbb{R}^{2n}} \alpha_0 \left\| \widetilde{\nabla}\mathbf{u} - \mathbf{w} \right\|_{2,1|\mathbb{R}^{2n}} + \alpha_1 \left\| \widetilde{\mathcal{E}}\mathbf{w} \right\|_{2,1|\mathbb{R}^{4n}}, \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^{2n}$, $\widetilde{\nabla} \in \mathbb{R}^{2n\times n}$ and $\widetilde{\mathcal{E}} \in \mathbb{R}^{4n\times 2n}$ are the discrete directional gradient operator and the directional symmetrized derivative, respectively, and $\alpha_0, \alpha_1 \in (0, +\infty)$. For any vector $\mathbf{v} \in \mathbb{R}^{2n}$ we set

$$\|\mathbf{v}\|_{2,1|\mathbb{R}^{2n}} = \sum_{j=1}^{n} \sqrt{v_j^2 + v_{n+j}^2}, \tag{4}$$

and for any vector $\mathbf{y} \in \mathbb{R}^{4n}$ we set

$$\|\mathbf{y}\|_{2,1|\mathbb{R}^{4n}} = \sum_{j=1}^{n} \sqrt{y_j^2 + y_{n+j}^2 + y_{2n+j}^2 + y_{3n+j}^2}. \tag{5}$$

Given an angle $\theta \in [-\pi, \pi]$ and a scaling parameter $a > 0$, we have that the discrete directional gradient operator has the form

$$\widetilde{\nabla} = \begin{bmatrix} D_\theta \\ D_{\theta^\perp} \end{bmatrix} = \begin{bmatrix} \cos(\theta)D_H + \sin(\theta)D_V \\ a(-\sin(\theta)D_H + \cos(\theta)D_V) \end{bmatrix},$$

where $D_\theta, D_{\theta^\perp} \in \mathbb{R}^{n\times n}$ represent the forward finite-difference operators along the directions determined by $\theta$ and $\theta^\perp = \theta + \frac{\pi}{2}$, respectively, and $D_H, D_V \in \mathbb{R}^{n\times n}$ represent the forward finite-difference operators along the horizontal and the vertical direction, respectively. Moreover, the directional symmetrized derivative is defined in block-wise form as

$$\widetilde{\mathcal{E}} = \begin{bmatrix} D_\theta & 0 \\ \frac{1}{2}D_{\theta^\perp} & \frac{1}{2}D_\theta \\ \frac{1}{2}D_{\theta^\perp} & \frac{1}{2}D_\theta \\ 0 & D_{\theta^\perp} \end{bmatrix}.$$

It is worth noting that, by fixing $\theta = 0$ and $a = 1$, we have $D_\theta = D_H$ and $D_{\theta^\perp} = D_V$, and the operators $\widetilde{\nabla}$ and $\widetilde{\mathcal{E}}$ define the TGV$^2$ regularization [4].

We observe that the definition of both the matrix $A$ and the finite difference operators $D_H$ and $D_V$ depend on the choice of boundary conditions. We make the following assumption.

**Assumption 1.** *We assume that periodic boundary conditions are considered for $A$, $D_H$ and $D_V$. Therefore, those matrices are Block Circulant with Circulant Blocks (BCCB).*

In this work we focus on directional images affected by Poisson noise, with the aim of assessing the behaviour of DTGV in this case. Besides extending the use of DTGV to Poisson noise, we introduce a novel technique for estimating the main direction of the image, which appears to be more efficient than the techniques applied in [9,10]. We solve the resulting optimization problem by using a customized version of the Alternating Direction Method of Multipliers (ADMM). We note that all the ADMM subproblems can be solved exactly at a low cost, thanks also to the use of FFTs, and that the method has proven convergence. Finally, we show the effectiveness of our approach on a set of test images, corrupted by out-of-focus and Gaussian blurs and noise with different signal-to-noise ratios. In particular, the KL-DTGV model of our problem is described in Section 2 and the technique for estimating the main direction is presented in Section 3. A detailed description of the ADMM version used for the minimization is given in Section 4 and the results of the numerical experiments are discussed in Section 5. Conclusions are given in Section 6.

Throughout this work we denote matrices with uppercase lightface letters, vectors with lowercase boldface letters and scalars with lowercase lightface letters. All the vectors are column vectors. Given a vector $\mathbf{v}$, we use $v_i$ or $(\mathbf{v})_i$ to denote its $i$-th entry. We use $\mathbb{R}_+$ to indicate the set of real nonnegative numbers and $\|\cdot\|$ to indicate the two-norm. For brevity, given any vectors $\mathbf{v}$ and $\mathbf{w}$ we use the notation $(\mathbf{v}, \mathbf{w})$ instead of $[\mathbf{v}^\top \ \mathbf{w}^\top]^\top$. Likewise, given any scalars $v$ and $w$, we use $(v, w)$ to indicate the vector $[v \ \ w]^\top$. We also use the notation $([\mathbf{v}]_1, [\mathbf{v}]_2)$ to highlight the subvectors $[\mathbf{v}]_1$ and $[\mathbf{v}]_2$ forming the vector $\mathbf{v}$. Finally, by writing $\mathbf{v} > 0$ we mean that all the entries of $\mathbf{v}$ are nonnegative and at least one of them is positive.

## 2. The KL-DTGV$^2$ Model

We briefly describe the KL-DTGV$^2$ model for the restoration of directional images corrupted by Poisson noise. Let $\mathbf{b} \in \mathbb{R}^n$ be the observed image. We want to recover the original image by minimizing a combination of the KL divergence (2) and the DTGV$^2$ regularizer (3), i.e., by solving the optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{u},\mathbf{w}} & \lambda\, D_{KL}(A\mathbf{u} + \gamma, \mathbf{b}) + \alpha_0 \left\| \widetilde{\nabla} \mathbf{u} - \mathbf{w} \right\|_{2,1|\mathbb{R}^{2n}} + \alpha_1 \left\| \widetilde{\mathcal{E}} \mathbf{w} \right\|_{2,1|\mathbb{R}^{4n}} \\ \text{s.t.} & \mathbf{u} \geq 0, \end{array} \tag{6}$$

where $\mathbf{u} \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $\gamma, \mathbf{b} \in \mathbb{R}^n$, $\mathbf{w} \in \mathbb{R}^{2n}$, and $\widetilde{\nabla} \in \mathbb{R}^{2n \times n}$ and $\widetilde{\mathcal{E}} \in \mathbb{R}^{4n \times 2n}$ are the linear operators defining the DTGV$^2$ regularization. The parameters $\lambda \in (0, +\infty)$ and $\alpha_0, \alpha_1 \in (0, 1)$ determine the balance between the KL data fidelity term and the two components of the regularization term.

We note that problem (6) is a nonsmooth convex optimization problem because of the properties of the KL divergence (see, e.g., [11]) and the DTGV operator (see, e.g., [10]).

## 3. Efficient Estimation of the Image Direction

An essential ingredient in the DTGV regularization is the estimation of the angle $\theta$ representing the image texture direction. In [10], an estimation algorithm based on the one in [12] is proposed, whose basic idea is to compute a pixelwise direction estimate and then $\theta$ as the average of that estimate. In [9], which focuses on impulse noise removal, a more efficient and robust algorithm for estimating the direction is presented, based on the Fourier transform. The main idea behind this algorithm is to exploit the fact that two-dimensional Fourier basis functions can be seen as images with one-directional patterns. However, despite being very efficient from a computational viewpoint, this technique does not appear to be fully reliable in our tests on Poissonian images (see Section 5.1). Therefore, we propose a different approach for estimating the direction, based on classical tools of image processing: the Sobel filter [13] and the Hough transform [14,15].

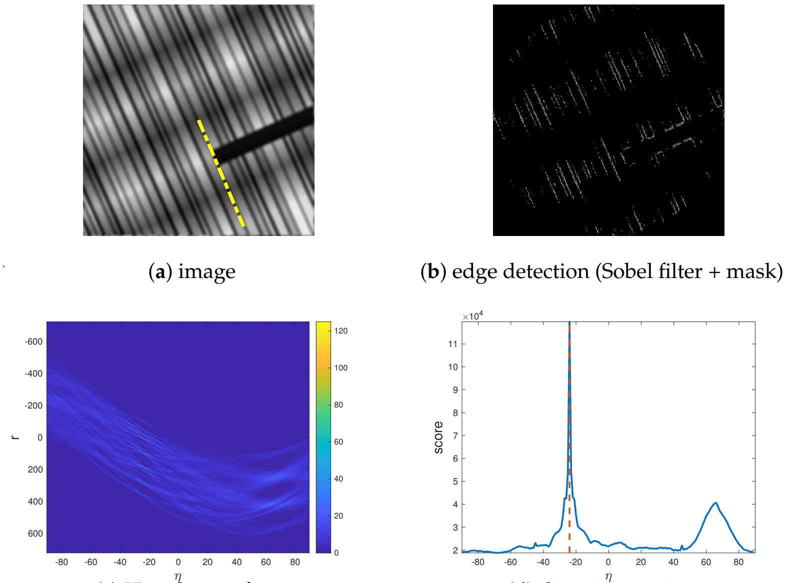
Our technique is based on the idea that if an image has a one-directional structure, i.e., its main pattern consists of stripes, then the edges of the image mainly consist of lines going in the direction of the stripes. The first stage of the proposed algorithm uses the Sobel filter to determine the edges of the noisy and blurry image. Then, the Hough transform is applied to the edge image in order to detect the lines. The Hough transform is based on the idea that each straight line can be identified by a pair $(r, \eta)$ where $r$ is the distance of the line from the origin, and $\eta$ is the angle between the $x$ axis and the segment connecting the origin with its orthogonal projection on the line. The output of the transform is a matrix in which each entry is associated with a pair $(r, \eta)$, i.e., with a straight line in the image, and its value is the sum of the values in the pixels that are on the line. Hence, the elements with the highest value in the Hough transform indicate the lines that are most likely to be present in the input image. Because of its definition, the Hough transform tends to overestimate diagonal lines in rectangular images (diagonal lines through the central part of the image contain the largest number of pixels); therefore, before computing the transform we apply a mask to the edge image, considering only the pixels inside the largest circle centered in the center of the image. After the Hough transform has been applied, we compute the square of the two-norm of each column of the matrix resulting from the transform, to determine a score for each angle from $-90°$ to $90°$. Intuitively, the score for each angle is related to the number of lines with that particular inclination which have been detected in the image. Finally, we set the direction estimate $\theta \in [-\pi, \pi]$ as

$$\theta = \begin{cases} \dfrac{90 - \eta_{\max}}{180}\pi, & \eta_{\max} \geq 0, \\ \dfrac{-90 - \eta_{\max}}{180}\pi, & \eta_{\max} < 0. \end{cases}$$

where $\eta_{\max}$ is the value of $\eta$ corresponding to the maximum score. A pseudocode for the estimation algorithm is provided in Algorithm 1 and an example of the algorithm workflow is given in Figure 1.

**Algorithm 1** Direction estimation.

```
Use the Sobel operator to obtain the image $\mathbf{e}$ of the edges of the noisy and blurry image $\mathbf{b}$.
Apply a disk mask to cut out some diagonal edges in $\mathbf{e}$, obtaining a new edge image $\widetilde{\mathbf{e}}$ (Figure 1b).
Compute the Hough transform $h(\widetilde{\mathbf{e}})$ (Figure 1c).
Set $\eta_{\max}$ as the value of $\eta$ corresponding to the column of $h(\widetilde{\mathbf{e}})$ with maximum 2-norm. (Figure 1d)
Set $\theta = \begin{cases} \frac{90-\eta_{\max}}{180}\pi, & \eta_{\max} \geq 0, \\ \frac{-90-\eta_{\max}}{180}\pi, & \eta_{\max} < 0. \end{cases}$ (yellow line in Figure 1a)
```



(**a**) image (**b**) edge detection (Sobel filter + mask)

(**c**) Hough transform (**d**) direction scoring

**Figure 1.** Workflow of Algorithm 1 on a random directional image.

## 4. ADMM for Minimizing the KL-DTGV$^2$ Model

Although problem (6) is a bound-constrained convex optimization problem, the nondifferentiability of the DTGV$^2$ regularizer does not allow its solution by classical optimization methods for smooth problems, such as gradient methods (see [16–18] and the references therein). However, the problem can be solved by methods based on splitting techniques, such as [19–23]. Here we solve (6) by the Alternating Direction Method of Multipliers (ADMM) [20]. To this end, we first reformulate the problem as follows:

$$\begin{array}{ll} \min\limits_{\mathbf{u},\mathbf{w},\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{z}_4} & \lambda\, D_{KL}(\mathbf{z}_1+\gamma,\mathbf{b}) + \alpha_0\, \|\mathbf{z}_2\|_{2,1|\mathbb{R}^{2n}} + \alpha_1\, \|\mathbf{z}_3\|_{2,1|\mathbb{R}^{4n}} + \chi_{\mathbb{R}^n_+}(\mathbf{z}_4) \\ \text{s.t.} & \mathbf{z}_1 = A\,\mathbf{u}, \\ & \mathbf{z}_2 = \widetilde{\nabla}\mathbf{u} - \mathbf{w}, \\ & \mathbf{z}_3 = \widetilde{\mathcal{E}}\mathbf{w}, \\ & \mathbf{z}_4 = \mathbf{u}, \end{array} \tag{7}$$

where $\mathbf{z}_1 \in \mathbb{R}^n$, $\mathbf{z}_2 \in \mathbb{R}^{2n}$, $\mathbf{z}_3 \in \mathbb{R}^{4n}$, $\mathbf{z}_4 \in \mathbb{R}^n$, and $\chi_{\mathbb{R}^n_+}(\mathbf{z}_4)$ is the characteristic function of the nonnegative orthant in $\mathbb{R}^n$. A similar splitting has been used in [24] for TV-based deblurring of Poissonian images. By introducing the auxiliary variables $\mathbf{x} = (\mathbf{u},\mathbf{w})$ and $\mathbf{z} = (\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{z}_4)$ we can further reformulate the KL-DTGV$^2$ problem as

$$\begin{array}{ll} \min\limits_{\mathbf{x},\mathbf{z}} & F_1(\mathbf{x}) + F_2(\mathbf{z}) \\ \text{s.t.} & H\,\mathbf{x} + G\,\mathbf{z} = 0, \end{array} \tag{8}$$

where we set

$$F_1(\mathbf{x}) = 0, \quad F_2(\mathbf{z}) = \lambda\, D_{KL}(\mathbf{z}_1+\gamma,\mathbf{b}) + \alpha_0\, \|\mathbf{z}_2\|_{2,1|\mathbb{R}^{2n}} + \alpha_1\, \|\mathbf{z}_3\|_{2,1|\mathbb{R}^{4n}} + \chi_{\mathbb{R}^n_+}(\mathbf{z}_4), \tag{9}$$

and we define the matrices $H \in \mathbb{R}^{8n\times 3n}$ and $G \in \mathbb{R}^{8n\times 8n}$ as

$$H = \begin{bmatrix} A & 0 \\ \widetilde{\nabla} & -I_{2n} \\ 0 & \widetilde{\mathcal{E}} \\ I_n & 0 \end{bmatrix}, \quad G = \begin{bmatrix} -I_n & 0 & 0 & 0 \\ 0 & -I_{2n} & 0 & 0 \\ 0 & 0 & -I_{4n} & 0 \\ 0 & 0 & 0 & -I_n \end{bmatrix}. \tag{10}$$

We consider the Lagrangian function associated with problem (8),

$$\mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\xi}) = F_1(\mathbf{x}) + F_2(\mathbf{z}) + \boldsymbol{\xi}^\top (H\,\mathbf{x} + G\,\mathbf{z}), \tag{11}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{8n}$ is a vector of Lagrange multipliers, and then the augmented Lagrangian function

$$\mathcal{L}_A(\mathbf{x},\mathbf{z},\boldsymbol{\xi};\rho) = F_1(\mathbf{x}) + F_2(\mathbf{z}) + \boldsymbol{\xi}^\top (H\,\mathbf{x} + G\,\mathbf{z}) + \frac{\rho}{2}\|H\,\mathbf{x} + G\,\mathbf{z}\|_2^2, \tag{12}$$

where $\rho > 0$.

Now we are ready to introduce the ADMM method for the solution of problem (8). Let $\mathbf{x}^0 \in \mathbb{R}^{3n}$, $\mathbf{z}^0 \in \mathbb{R}^{8n}$, $\boldsymbol{\xi}^0 \in \mathbb{R}^{8n}$. At each step $k > 0$ the ADMM method computes the new iterate $\left(\mathbf{x}^{k+1},\mathbf{z}^{k+1},\boldsymbol{\xi}^{k+1}\right)$ as follows:

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}\in\mathbb{R}^{3n}} \mathcal{L}_A(\mathbf{x},\mathbf{z}^k,\boldsymbol{\xi}^k;\rho), \\ \mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}\in\mathbb{R}^{8n}} \mathcal{L}_A(\mathbf{x}^{k+1},\mathbf{z},\boldsymbol{\xi}^k;\rho), \\ \boldsymbol{\xi}^{k+1} &= \boldsymbol{\xi}^k + \rho\left(H\,\mathbf{x}^{k+1} + G\,\mathbf{z}^{k+1}\right). \end{aligned} \tag{13}$$

Note that the functions $F_1(\mathbf{x})$ and $F_2(\mathbf{z})$ in (8) are closed, proper and convex. Moreover, the matrices $H$ and $G$ defined in (10) are such that $G = -I_{8n}$ and $H$ has full rank. Hence, the convergence of the method defined by (13) can be proved by applying a classical convergence result from the seminal paper by Eckstein and Bertsekas [25] (Theorem 8), which we report in a form that can be immediately applied to our reformulation of the problem.

**Theorem 1.** *Let us consider a problem of the form (8) where $F_1(\mathbf{x})$ and $F_2(\mathbf{z})$ are closed, proper and convex functions and $H$ has full rank. Let $\mathbf{x}^0 \in \mathbb{R}^{3n}$, $\mathbf{z}^0 \in \mathbb{R}^{8n}$, $\boldsymbol{\xi}^0 \in \mathbb{R}^{8n}$, and $\rho > 0$. Suppose $\{\varepsilon_k\}, \{\nu_k\} \subset \mathbb{R}_+$ are summable sequences such that for all $k$*

$$\begin{aligned}
&\left\| \mathbf{x}^{k+1} - \arg\min_{\mathbf{x}\in\mathbb{R}^{3n}} \mathcal{L}_A(\mathbf{x}, \mathbf{z}^k, \boldsymbol{\xi}^k; \rho) \right\| \le \varepsilon_k, \\
&\left\| \mathbf{z}^{k+1} - \arg\min_{\mathbf{z}\in\mathbb{R}^{8n}} \mathcal{L}_A(\mathbf{x}^{k+1}, \mathbf{z}, \boldsymbol{\xi}^k; \rho) \right\| \le \nu_k, \\
&\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k + \rho\left(H\,\mathbf{x}^{k+1} + G\,\mathbf{z}^{k+1}\right).
\end{aligned}$$

*If there exists a saddle point $(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\xi}^*)$ of $\mathcal{L}(\mathbf{x}, \mathbf{z}, \boldsymbol{\xi})$, then $\mathbf{x}^k \to \mathbf{x}^*$, $\mathbf{z}^k \to \mathbf{z}^*$ and $\boldsymbol{\xi}^k \to \boldsymbol{\xi}^*$. If such saddle point does not exist, then at least one of the sequences $\{\mathbf{z}^k\}$ or $\{\boldsymbol{\xi}^k\}$ is unbounded.*

Since we are dealing with linear constraints, we can recast (13) in a more convenient form, by observing that the linear term in (12) can be included in the quadratic one. By introducing the vector of scaled Lagrange multipliers $\boldsymbol{\mu}^k = \frac{1}{\rho}\boldsymbol{\xi}^k$, the ADMM method becomes

$$\begin{aligned}
\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}\in\mathbb{R}^{3n}} \frac{\rho}{2}\left\| H\,\mathbf{x} - \mathbf{z}^k + \boldsymbol{\mu}^k \right\|_2^2, && (14) \\
\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}\in\mathbb{R}^{8n}} F_2(\mathbf{z}) + \frac{\rho}{2}\left\| H\,\mathbf{x}^{k+1} - \mathbf{z} + \boldsymbol{\mu}^k \right\|_2^2, && (15) \\
\boldsymbol{\mu}^{k+1} &= \boldsymbol{\mu}^k + H\,\mathbf{x}^{k+1} + G\,\mathbf{z}^{k+1}. && (16)
\end{aligned}$$

In the next sections we show how the solutions to subproblems (14) and (15) can be computed exactly with a small computational effort.

### 4.1. Solving the Subproblem in x

Problem (14) is an overdetermined least squares problem, since $H$ is a tall-and-skinny matrix with full rank. Hence, its solution can be computed by solving the normal equations system

$$H^\top H\,\mathbf{x} = H^\top \mathbf{v}_x^k, \tag{17}$$

where we set $\mathbf{v}_x^k = \mathbf{z}^k - \boldsymbol{\mu}^k$. Starting from the definition of $H$ given in (10), we have

$$\begin{aligned}
H^\top H &= \left[\begin{array}{c|c} I_n + A^\top A + \widetilde{\nabla}^\top\widetilde{\nabla} & -\widetilde{\nabla}^\top \\ \hline -\widetilde{\nabla} & I_{2n} + \widetilde{\mathcal{E}}^\top\widetilde{\mathcal{E}} \end{array}\right] = \\
&= \left[\begin{array}{c|cc} I_n + A^\top A + \widetilde{\nabla}^\top\widetilde{\nabla} & -D_\theta^\top & -D_{\theta^\perp}^\top \\ \hline -D_\theta & I_n + D_\theta^\top D_\theta + \frac{1}{2}D_{\theta^\perp}^\top D_{\theta^\perp} & \frac{1}{2}D_{\theta^\perp}^\top D_\theta \\ -D_{\theta^\perp} & \frac{1}{2}D_\theta^\top D_{\theta^\perp} & I_n + \frac{1}{2}D_\theta^\top D_\theta + D_{\theta^\perp}^\top D_{\theta^\perp} \end{array}\right].
\end{aligned}$$

System (17) may be quite large and expensive, also for relatively small images. However, as pointed out in Assumption 1, $A$, $D_\theta$ and $D_{\theta^\perp}$ have a BCCB structure, hence all the blocks of $H^\top H$ maintain that structure. By recalling that BCCB matrices can be diagonalized by means of two-dimensional Discrete Fourier Transforms (DFTs), we show how the solution to (17) can be computed expeditiously.

Let $\mathcal{F} \in \mathbb{C}^{n\times n}$ be the matrix representing the two-dimensional DFT operator, and let $\mathcal{F}^*$ denote its inverse, i.e., its adjoint. We can write $H^\top H$ as

$$H^\top H = \left[\begin{array}{ccc} \mathcal{F}^* & 0 & 0 \\ 0 & \mathcal{F}^* & 0 \\ 0 & 0 & \mathcal{F}^* \end{array}\right] \left[\begin{array}{c|cc} \Gamma & -\Delta_\theta^* & -\Delta_{\theta^\perp}^* \\ \hline -\Delta_\theta & \Phi_{11} & \Phi_{12} \\ -\Delta_{\theta^\perp} & \Phi_{21} & \Phi_{22} \end{array}\right] \left[\begin{array}{ccc} \mathcal{F} & 0 & 0 \\ 0 & \mathcal{F} & 0 \\ 0 & 0 & \mathcal{F} \end{array}\right], \tag{18}$$

where each block of the central matrix is the diagonal complex matrix associated with the corresponding block in $H^\top H$, and $\Delta_\theta^*, \Delta_{\theta^\perp}^*$ denote the (diagonal) adjoint matrices of $\Delta_\theta, \Delta_{\theta^\perp}$. By (18) and the definition of $\mathbf{x}$, we can reformulate (17) as

$$\left[\begin{array}{c|cc} \Gamma & -\Delta_\theta^* & -\Delta_{\theta^\perp}^* \\ \hline -\Delta_\theta & \Phi_{11} & \Phi_{12} \\ -\Delta_{\theta^\perp} & \Phi_{21} & \Phi_{22} \end{array}\right] \left[\begin{array}{c} \mathcal{F}\,\mathbf{u} \\ \mathcal{F}\,\mathbf{w}_1 \\ \mathcal{F}\,\mathbf{w}_2 \end{array}\right] = \left[\begin{array}{c} \mathcal{F}\,[H^\top \mathbf{v}_x^k]_1 \\ \mathcal{F}\,[H^\top \mathbf{v}_x^k]_2 \\ \mathcal{F}\,[H^\top \mathbf{v}_x^k]_3 \end{array}\right], \tag{19}$$

where we split $\mathbf{w}$ and $\mathbf{v}_x^k$ in two and three blocks of size $n$, respectively.

Now we recall a result about the inversion of block matrices. Suppose that a square matrix $M$ is partitioned into four blocks, i.e.,

$$M = \left[\begin{array}{cc} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array}\right];$$

then, if $M_{11}$ and $M_{22}$ are invertible, we have

$$\begin{aligned} M^{-1} &= \left[\begin{array}{cc} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array}\right]^{-1} \\ &= \left[\begin{array}{cc} \left(M_{11} - M_{12}M_{22}^{-1}M_{21}\right)^{-1} & 0 \\ 0 & \left(M_{22} - M_{21}M_{11}^{-1}M_{12}\right)^{-1} \end{array}\right] \left[\begin{array}{cc} I & -M_{12}M_{22}^{-1} \\ -M_{21}M_{11}^{-1} & I \end{array}\right]. \end{aligned} \tag{20}$$

By applying (20) to the matrix consisting of the second and third block rows and columns of the matrix in (19), which we denote $\Phi$, we get

$$\begin{aligned} \Phi^{-1} &= \left[\begin{array}{cc} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{array}\right]^{-1} \\ &= \left[\begin{array}{cc} \left(\Phi_{11} - \Phi_{12}\Phi_{22}^{-1}\Phi_{21}\right)^{-1} & -\left(\Phi_{11} - \Phi_{12}\Phi_{22}^{-1}\Phi_{21}\right)^{-1}\Phi_{12}\Phi_{22}^{-1} \\ -\left(\Phi_{22} - \Phi_{21}\Phi_{11}^{-1}\Phi_{12}\right)^{-1}\Phi_{21}\Phi_{11}^{-1} & \left(\Phi_{22} - \Phi_{21}\Phi_{11}^{-1}\Phi_{12}\right)^{-1} \end{array}\right]. \end{aligned} \tag{21}$$

To simplify the notation we set

$$\Psi = \left[\begin{array}{cc} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{array}\right] = \Phi^{-1}, \tag{22}$$

and observe that the matrices $\Psi_{ij} \in \mathbb{C}^{n\times n}$ are diagonal. Letting $\Delta^* = \left[\Delta_\theta^* \;\; \Delta_{\theta^\perp}^*\right]$, applying the inversion formula (20) to the whole matrix in (19), and using (21) and (22), we get

$$\left[\begin{array}{cc} \Gamma & -\Delta^* \\ -\Delta & \Phi \end{array}\right]^{-1} = \left[\begin{array}{cc} \Xi^{-1} & 0 \\ 0 & \Omega^{-1} \end{array}\right] \left[\begin{array}{cc} I_n & -\Delta^*\Psi \\ -\Delta\,\Gamma^{-1} & I_{2n} \end{array}\right], \tag{23}$$

where

$$\begin{aligned} \Xi &= \Gamma - \Delta^*\Psi\,\Delta = \Gamma - \left[\Delta_\theta^* \;\; \Delta_{\theta^\perp}^*\right] \left[\begin{array}{cc} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{array}\right] \left[\begin{array}{c} \Delta_\theta \\ \Delta_{\theta^*} \end{array}\right], \\ \Omega &= \Phi - \Delta\,\Gamma^{-1}\Delta^* = \left[\begin{array}{cc} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{array}\right] - \left[\begin{array}{c} \Delta_\theta \\ \Delta_{\theta^\perp} \end{array}\right] \Gamma^{-1} \left[\Delta_\theta^* \;\; \Delta_{\theta^\perp}^*\right]. \end{aligned}$$

We note that $\Xi \in \mathbb{C}^{n\times n}$ is diagonal (and its inversion is straightforward), while $\Omega \in \mathbb{C}^{2n\times 2n}$ has a $2 \times 2$ block structure with blocks that are diagonal matrices belonging to $\mathbb{C}^{n\times n}$. Thus, we can compute $\Upsilon = \Omega^{-1}$ by applying (20):

$$\begin{aligned} \Upsilon = \begin{bmatrix} \Upsilon_{11} & \Upsilon_{12} \\ \Upsilon_{21} & \Upsilon_{22} \end{bmatrix} &= \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \left(\Omega_{11} - \Omega_{12}\Omega_{22}^{-1}\Omega_{21}\right)^{-1} & -\left(\Omega_{11} - \Omega_{12}\Omega_{22}^{-1}\Omega_{21}\right)^{-1}\Omega_{12}\Omega_{22}^{-1} \\ -\left(\Omega_{22} - \Omega_{21}\Omega_{11}^{-1}\Omega_{12}\right)^{-1}\Omega_{21}\Omega_{11}^{-1} & \left(\Omega_{22} - \Omega_{21}\Omega_{11}^{-1}\Omega_{12}\right)^{-1} \end{bmatrix} \end{aligned} \tag{24}$$

Summing up, by (19), (23) and (24), the solution to (17) can be obtained by computing

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} \Xi^{-1} & 0 \\ 0 & \Upsilon \end{bmatrix} \begin{bmatrix} I_n & -\Delta^\top \Psi \\ -\Delta\,\Gamma^{-1} & I_{2n} \end{bmatrix} \begin{bmatrix} \mathcal{F}\,[H^\top \mathbf{v}_x^k]_1 \\ \mathcal{F}\,[H^\top \mathbf{v}_x^k]_2 \\ \mathcal{F}\,[H^\top \mathbf{v}_x^k]_3 \end{bmatrix}, \tag{25}$$

and setting

$$\mathbf{u}^{k+1} = \mathcal{F}^* \mathbf{y}_1, \qquad \mathbf{w}_1^{k+1} = \mathcal{F}^* \mathbf{y}_2, \qquad \mathbf{w}_2^{k+1} = \mathcal{F}^* \mathbf{y}_3. \tag{26}$$

**Remark 1.** *The only quantity in (25) that varies at each iteration is $\mathbf{v}_x^k$. Hence, the matrices $\Delta$, $\Gamma$, $\Psi$, $\Xi^{-1}$, and $\Upsilon$ can be computed only once before the ADMM method starts. This means that the overall cost of the exact solution of (14) at each iteration reduces to six two-dimensional DFTs and two matrix–vector products involving two $3 \times 3$ block matrices with diagonal blocks of dimension $n$.*

### 4.2. Solving the Subproblem in $\mathbf{z}$

By looking at the form of $F_2(\mathbf{z})$–see (9)–and by defining the vector $\mathbf{v}_z^k = H\,\mathbf{x}^{k+1} + \boldsymbol{\mu}^k$, we see that problem (15) can be split into the four problems

$$\mathbf{z}_1^{k+1} = \arg\min_{\mathbf{z}_1 \in \mathbb{R}^n} \lambda D_{KL}(\mathbf{z}_1 + \boldsymbol{\gamma}, \mathbf{b}) + \frac{\rho}{2}\left\| \mathbf{z}_1 - [\mathbf{v}_z^k]_1 \right\|_2^2, \tag{27}$$

$$\mathbf{z}_2^{k+1} = \arg\min_{\mathbf{z}_2 \in \mathbb{R}^{2n}} \alpha_0 \|\mathbf{z}_2\|_{2,1|\mathbb{R}^{2n}} + \frac{\rho}{2}\left\| \mathbf{z}_2 - [\mathbf{v}_z^k]_2 \right\|_2^2, \tag{28}$$

$$\mathbf{z}_3^{k+1} = \arg\min_{\mathbf{z}_3 \in \mathbb{R}^{4n}} \alpha_1 \|\mathbf{z}_3\|_{2,1|\mathbb{R}^{4n}} + \frac{\rho}{2}\left\| \mathbf{z}_3 - [\mathbf{v}_z^k]_3 \right\|_2^2, \tag{29}$$

$$\mathbf{z}_4^{k+1} = \arg\min_{\mathbf{z}_4 \in \mathbb{R}^n} \chi_{\mathbb{R}_+^n}(\mathbf{z}_4) + \frac{\rho}{2}\left\| \mathbf{z}_4 - [\mathbf{v}_z^k]_4 \right\|_2^2, \tag{30}$$

where $\mathbf{v}_z^k = ([\mathbf{v}_z^k]_1, [\mathbf{v}_z^k]_2, [\mathbf{v}_z^k]_3, [\mathbf{v}_z^k]_4)$, with $[\mathbf{v}_z^k]_1 \in \mathbb{R}^n$, $[\mathbf{v}_z^k]_2 \in \mathbb{R}^{2n}$,$[\mathbf{v}_z^k]_3 \in \mathbb{R}^{4n}$, and $[\mathbf{v}_z^k]_4 \in \mathbb{R}^n$. Now we focus on the solution of the four subproblems.

#### 4.2.1. Update of $\mathbf{z}_1$

By the form of the Kullback–Leibler divergence in (2), the minimization problem (27) is equivalent to

$$\min_{\mathbf{z}_1 \in \mathbb{R}^n} \lambda \sum_{i=1}^n \left( b_i \ln \frac{b_i}{(\mathbf{z}_1)_i + \gamma_i} + (\mathbf{z}_1)_i + \gamma_i - b_i \right) + \frac{\rho}{2} \sum_{i=1}^n ((\mathbf{z}_1)_i - d_i)^2, \tag{31}$$

where we set $\mathbf{d} = [\mathbf{v}_z^k]_1$ to ease the notation. From (31) it is clear that the problem in $\mathbf{z}_1$ can be split into $n$ problems of the form

$$\min_{z \in \mathbb{R}} \lambda(-b \ln(z + \gamma) + z) + \frac{\rho}{2}(z - d)^2. \tag{32}$$

Since the objective function of this problem is strictly convex, its solution can be determined by setting the gradient equal to zero, i.e., by solving

$$\lambda\left(-\frac{b}{z+\gamma}+1\right)+\rho(z-d)=0,$$

which leads to the quadratic equation

$$z^2+\left(\frac{\lambda}{\rho}+\gamma-d\right)z-\frac{\lambda}{\rho}\left(\frac{\rho}{\lambda}\gamma d-\gamma+b\right)=0. \tag{33}$$

Since, by looking at the domain of the objective function in (32), $z+\gamma$ has to be strictly positive, we set each entry of $\mathbf{z}_1^{k+1}$ as the largest solution of the corresponding quadratic Equation (33).

#### 4.2.2. Update of $\mathbf{z}_2$ and $\mathbf{z}_3$

The minimization problems (28) and (29) correspond to the computation of the proximal operators of the functions $f(\mathbf{z}_2)=\frac{\alpha_0}{\rho}\|\mathbf{z}_2\|_{2,1|\mathbb{R}^{2n}}$ and $g(\mathbf{z}_3)=\frac{\alpha_1}{\rho}\|\mathbf{z}_3\|_{2,1|\mathbb{R}^{4n}}$, respectively.

By the definitions given in (4) and (5), we see that the two (2,1)-norms correspond to the sum of two-norms of vectors in $\mathbb{R}^2$ and $\mathbb{R}^4$, respectively. This means that the computation of both the proximal operators can be split into the computation of $n$ proximal operators of functions that are scaled two-norms in either $\mathbb{R}^2$ or $\mathbb{R}^4$.

The proximal operator of the function $f(\mathbf{y})=c\|\mathbf{y}\|$, $c>0$, at a vector $\mathbf{d}$ is

$$\mathrm{prox}_{c\|\cdot\|}(\mathbf{d})=\arg\min_{\mathbf{y}} c\|\mathbf{y}\|+\frac{1}{2}\|\mathbf{y}-\mathbf{d}\|^2.$$

It can be shown (see, e.g., [26] [Chapter 6]) that

$$\mathrm{prox}_{c\|\cdot\|}(\mathbf{d})=\left(1-\frac{c}{\max\{\|\mathbf{d}\|,\,c\}}\right)\mathbf{d}=\max\left\{\frac{\|\mathbf{d}\|-c}{\|\mathbf{d}\|},\,0\right\}\mathbf{d}. \tag{34}$$

Hence, for the update of $\mathbf{z}_2$ we proceed as follows. By setting $\mathbf{d}=[\mathbf{v}_z^k]_2$ and $c=\frac{\alpha_0}{\rho}$, for each $i=1,\ldots,n$ we have

$$\left((\mathbf{z}_2^{k+1})_i,\,(\mathbf{z}_2^{k+1})_{n+i}\right)=\mathrm{prox}_{c\|\cdot\|}((d_i,\,d_{n+i})).$$

To update $\mathbf{z}_3$, we set $\mathbf{d}=[\mathbf{v}_z^k]_3$ and $c=\frac{\alpha_1}{\rho}$ and compute

$$\left((\mathbf{z}_3^{k+1})_i,\,(\mathbf{z}_3^{k+1})_{n+i},\,(\mathbf{z}_3^{k+1})_{2n+i},\,(\mathbf{z}_3^{k+1})_{3n+i}\right)=\mathrm{prox}_{c\|\cdot\|}((d_i,\,d_{n+i},\,d_{2n+i},\,d_{3n+i})).$$

#### 4.2.3. Update of $\mathbf{z}_4$

It is straightforward to verify that the update of $\mathbf{z}_4$ in (30) can be obtained as

$$\mathbf{z}_4^{k+1}=\Pi_{\mathbb{R}^n_+}\left([\mathbf{v}_z^k]_4\right),$$

where $\Pi_{\mathbb{R}^n_+}$ is the Euclidean projection onto the nonnegative orthant in $\mathbb{R}^n$.

### *4.3. Summary of the ADMM Method*

For the sake of clarity, in Algorithm 2 we sketch the ADMM version for solving problem (7).

In many image restoration applications, a reasonably good starting guess for $\mathbf{u}$ is often available. For example, if $A$ represents a blur operator, a common choice is to set $\mathbf{u}^0$ equal to the the noisy and blurry image. We make this choice for $\mathbf{u}^0$. By numerical experiments we also verified that once $\mathbf{x}=(\mathbf{u},\mathbf{w})$ has been initialized, it is convenient to set $\mathbf{u}^1=\mathbf{u}^0$, $\mathbf{w}_1^1=\mathbf{w}_1^0$ and $\mathbf{w}_2^1=\mathbf{w}_2^0$ and to shift the order of the updates in the ADMM scheme (14)–(16), so that a "more effective" initialization of $\mathbf{z}$ and $\boldsymbol{\mu}$ is performed. We see from line 9 of Algorithm 2 that the algorithm stops when the relative change in the restored image $\mathbf{u}$ goes below a certain threshold $tol \in (0,1)$ or a maximum number of iterations $k_{\max}$ is reached. Finally, we note that for the case of the KL-TGV$^2$ model, corresponding to $\theta = 0$ and $a = 1$, we have that $D_\theta = D_H$ and $D_{\theta^\perp} = D_V$; hence, we use the initialization $\mathbf{w}_1^0 = D_H\mathbf{u}^0$ and $\mathbf{w}_2^0 = D_V\mathbf{u}^0$.

**Algorithm 2** ADMM for problem (7).

```
Let $\mathbf{u}^0 \in \mathbb{R}^n$, $\mathbf{w}_1^0 = D_\theta \mathbf{u}^0$, $\mathbf{w}_2^0 = D_{\theta^\perp}\mathbf{u}^0$, $\boldsymbol{\mu}^0 = 0$, $\mathbf{z}^0 = 0$, $\lambda, \rho \in (0, +\infty)$, $\alpha_0, \alpha_1 \in (0,1)$
Compute matrices $\Delta$, $\Gamma$, $\Psi$, $\Xi^{-1}$, and $\Upsilon$ as specified in Section 4.1
Let $k = 0$, $\mathbf{u}^1 = \mathbf{u}^0$, $\mathbf{w}^1 = \mathbf{w}^0$, $stop = false$, $tol \in (0,1)$, $k_{\max} \in \mathbb{N}$
while not $stop$ and $k \le k_{\max}$ do
    Compute $\mathbf{z}^{k+1}$ by solving the four subproblems (27)–(30)
    Compute $\boldsymbol{\mu}^{k+1}$ as in (16)
    $k = k + 1$
    Compute $\mathbf{u}^{k+1}$, $\mathbf{w}_1^{k+1}$ and $\mathbf{w}_2^{k+1}$ by (25) and (26)
    Set $stop = \left(\|\mathbf{u}^{k+1} - \mathbf{u}^k\| < tol\,\|\mathbf{u}^k\|\right)$
end while
```

## 5. Numerical Results

All the experiments were carried out using MATLAB R2018a on a 3.50 GHz Intel Xeon E3 with 16 GB of RAM and Windows operating system. In this section, we first illustrate the effectiveness of Algorithm 1 for the estimation of the image direction by comparing it with the one given in [9] and by analysing its sensitivity to the degradation in the image to be restored. Then, we present numerical experiments that demonstrate the improvement of the KL-DTGV$^2$ model upon the KL-TGV$^2$ model for the restoration of directional images corrupted by Poisson noise.

Four directional images named `phantom` (512 × 512), `grass` (375 × 600), `leaves` (203 × 300) and `carbon` (247 × 300) were used in the experiments. The first image is a piecewise affine fibre phantom image obtained with the `fibre_phantom_pa` MATLAB function available from http://www2.compute.dtu.dk/~pcha/HDtomo/ (accessed on 20 September 2020). The second and third images represent grass and veins of leaves, respectively, which naturally exhibit a directional structure. The last image is a Scanning Electron Microscope (SEM) image of carbon fibres. The images are shown in Figures 2–5.

To simulate experimental data, each reference image was convolved with two PSFs, one corresponding to a Gaussian blur with variance 2, generated by the `psfGauss` function from [27], and the other corresponding to an out-of-focus blur with radius 5, obtained with the function `fspecial` from the MATLAB Image Processing Toolbox. To take into account the existence of some background emission, a constant term $\gamma$ equal to $10^{-10}$ was added to all pixels of the blurry image. The resulting image was corrupted by Poisson noise, using the MATLAB function `imnoise`. The intensities of the original images were pre-scaled to get noisy and blurry images with Signal to Noise Ratio (SNR) equal to 43 and 37 dB. We recall that in the case of Poisson noise, which affects the photon counting process, the SNR is estimated as [28]

$$\mathrm{SNR} = 10\log_{10}\left(\frac{N_{\mathrm{exact}}}{\sqrt{N_{\mathrm{exact}} + N_{\mathrm{background}}}}\right),$$

where $N_{\mathrm{exact}}$ and $N_{\mathrm{background}}$ are the total number of photons in the image to be recovered and in the background term, respectively. Finally, the corrupted images were scaled to have their maximum intensity values equal to 1. For each test problem, the noisy and blurry images are shown in Figures 2–5.

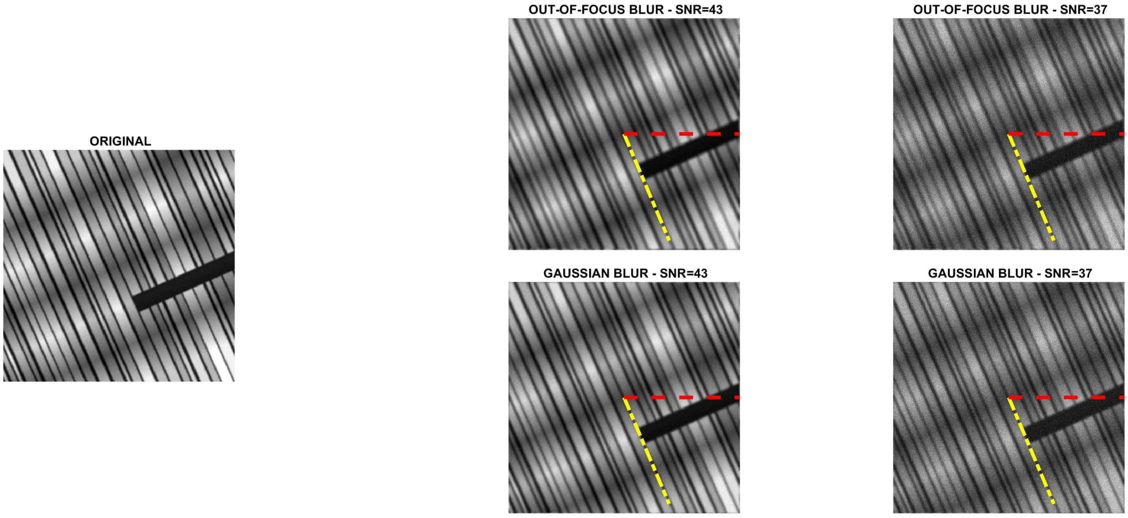


**Figure 2.** Test problem `phantom`: original and corrupted images. The yellow dash-dotted line indicates the direction estimated by Algorithm 1 and the red dashed line the direction estimated by the method in [9].

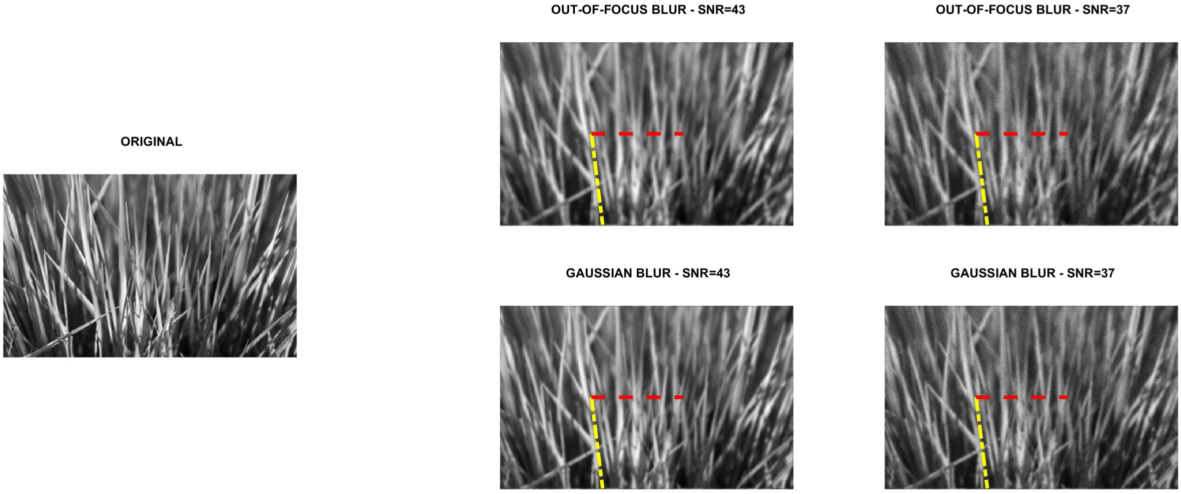


**Figure 3.** Test problem `grass`: original and corrupted images. The yellow dash-dotted line indicates the direction estimated by Algorithm 1 and the red dashed line the direction estimated by the method in [9].

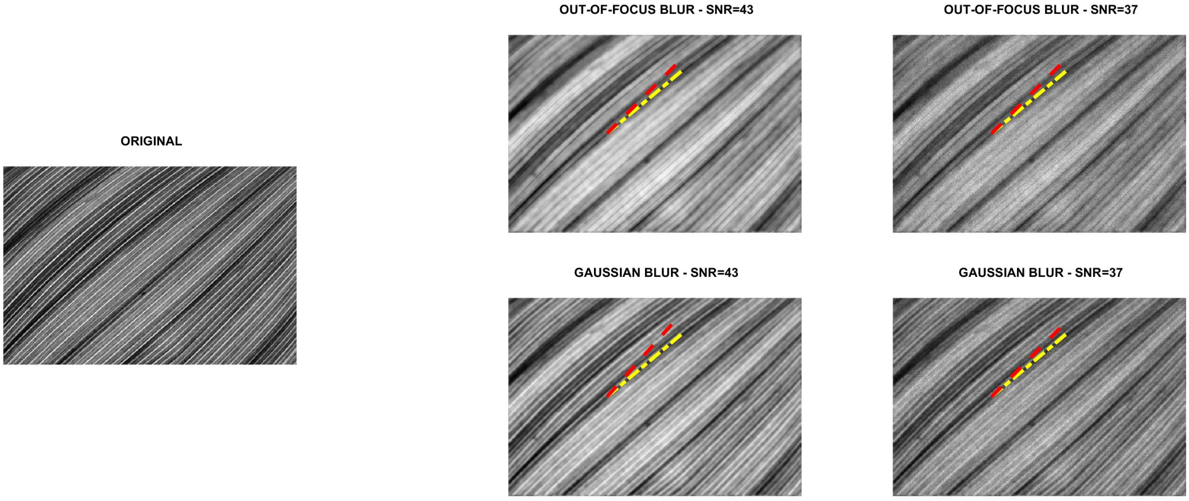


**Figure 4.** Test problem `leaves`: original and corrupted images. The yellow dash-dotted line indicates the direction estimated by Algorithm 1 and the red dashed line the direction estimated by the method in [9].

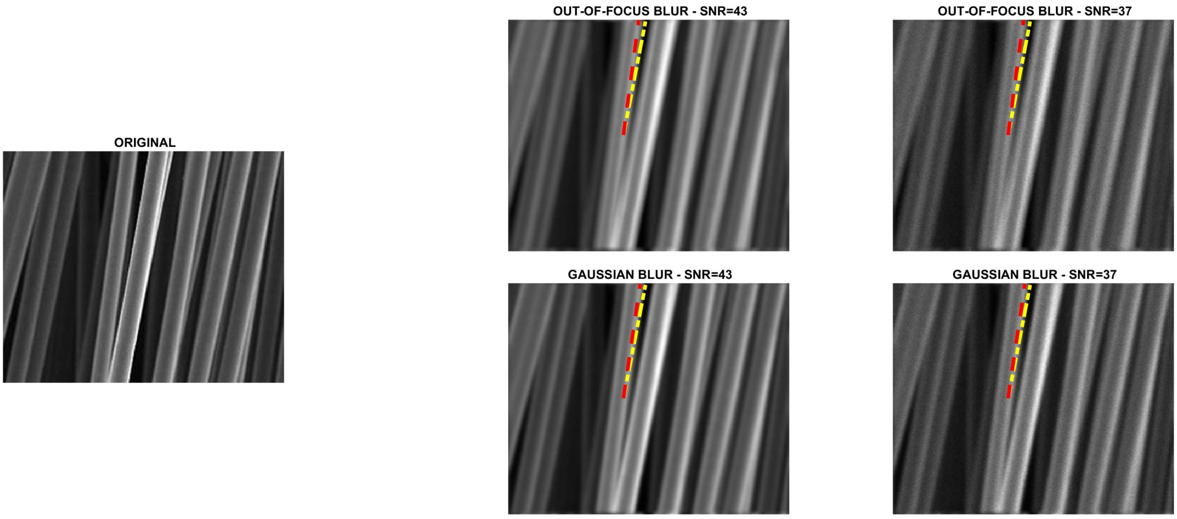


**Figure 5.** Test problem `carbon`: original and corrupted images. The yellow dash-dotted line indicates the direction estimated by Algorithm 1 and the red dashed line the direction estimated by the method in [9].

### *5.1. Direction Estimation*

In Figures 2–5 we compare Algorithm 1 with the algorithm proposed in [9], showing that Algorithm 1 always correctly estimates the main direction of the four test images. We also test the robustness of our algorithm with respect to noise and blur. In Figure 6 we show the estimated main direction of the `phantom` image corrupted by Poisson noise with SNR $= 35, 37, 39, 41, 43$ dB and out-of-focus blurs with radius R $= 5, 7, 9$. In only one case (SNR $= 35$, R $= 7$) Algorithm 1 fails, returning as estimate the orthogonal direction, i.e., the direction corresponding to the large black line and the background color gradient. Finally, we test Algorithm 1 on a phantom image with vertical, horizontal and diagonal main directions corresponding to $\theta = 0, 90, 45$. The results, in Figure 7, show that our algorithm is not sensitive to the specific directional structure of the image.

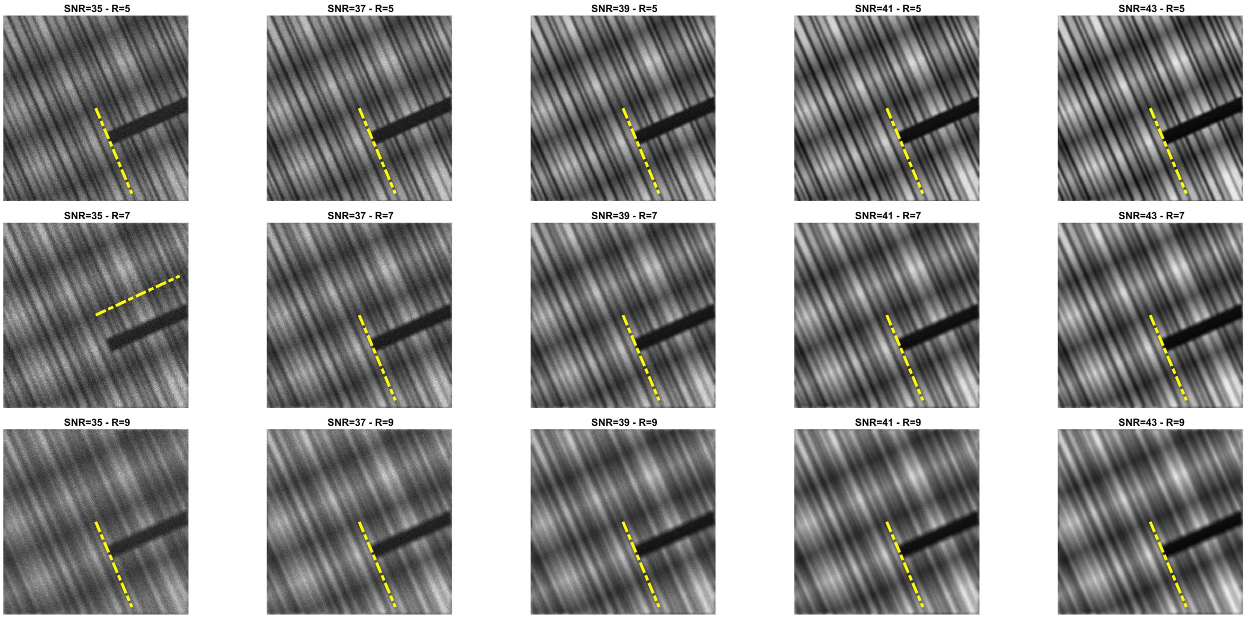


**Figure 6.** Direction estimation for `phantom` with SNR $= 35, 37, 39, 41, 43$ dB (from left to right) and out-of-focus blur with radius R $= 5, 7, 9$ (from top to bottom).

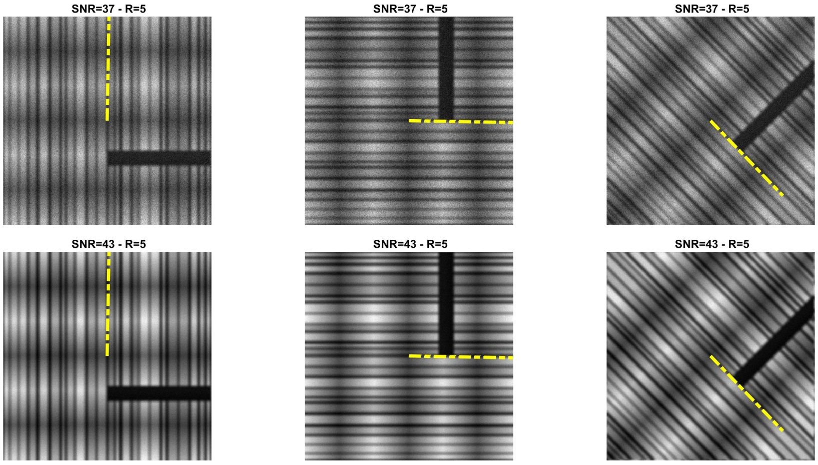


**Figure 7.** Direction estimation for `phantom` with SNR $= 37, 43$ (top, bottom) and out-of-focus blur with radius R $= 5$.

### 5.2. Image Deblurring

We compare the quality of the restorations obtained by using the DTGV$^2$ and TGV$^2$ regularizers and ADMM for the solution of both models. In all the tests, the value of the penalty parameter was set as $\rho = 10$ and the value of the stopping threshold as $tol = 10^{-4}$. A maximum number of $k_{\max} = 500$ iterations was allowed. By following [9,10], the weight parameters of DTGV were chosen as $\alpha_0 = \beta$ and $\alpha_1 = (1 - \beta)$ with $\beta = 2/3$. For each test problem, the value of the regularization parameter $\lambda$ was tuned by a trial-and-error strategy. This strategy consisted in running ADMM with initial guess $\mathbf{u}^0 = \mathbf{b}$ several times on each test image, varying the value of $\lambda$ at each execution. For all the runs the stopping criterion for ADMM and the values of $\alpha_0$, $\alpha_1$ and $\rho$ were the same as described above. The value of $\lambda$ yielding the smallest Root Mean Square Error (RMSE) at the last iteration was chosen as the "optimal" value.

The numerical results are summarized in Table 1, where the RMSE, the Improved Signal to Noise Ratio (ISNR) [29], and the structural similarity (SSIM) index [30] are used to give a quantitative evaluation of the quality of the restorations. As a measure of the computational cost, the number of iterations and the time in seconds are reported. Table 1 also shows, for each test problem, the values of the regularization parameter $\lambda$. The restored images are shown in Figures 8–11. For the `carbon` test problem, Figure 12 shows the error images, i.e., the images obtained as the absolute difference between the original image and the restored one. The values of the pixels of the error images have been scaled in the range $[m, M]$ where $m$ and $M$ are the minimum and maximum pixel value of the DTGV$^2$ and TGV$^2$ error images.

From the results, it is evident that the DTGV$^2$ model outperforms the TGV$^2$ one in terms of quality of the restoration. A visual inspection of the figures shows that the DTGV$^2$ regularization is very effective in removing the noise, while for high noise levels the TGV$^2$ reconstructions still exhibit noise artifacts. Finally, by observing the "Iters" column of the table, we can conclude that, on average, the TGV$^2$ regularization requires less ADMM iterations to achieve a relative change in the restoration that is below the fixed threshold. However, the computational time per iteration is very small and also ADMM for the KL-DGTV$^2$ regularization is efficient.

Finally, to illustrate the behaviour of ADMM, in Figure 13 we plot the RMSE history for the `carbon` test problem. A similar RMSE behaviour has been observed in all the numerical experiments.

**Table 1.** Numerical results for the test problems.

| Blur | SNR | Model | $\lambda$ | RMSE | ISNR | MSSIM | Iters | Time |
|---|---|---|---|---|---|---|---|---|
| `phantom` | | | | | | | | |
| Out-of-focus | 43 | DTGV | 57.5 | $2.2558 \times 10^{-2}$ | 9.5472 | $9.3007 \times 10^{-1}$ | 86 | 10.95 |
| | | TGV | 275 | $2.8043 \times 10^{-2}$ | 7.6568 | $8.9887 \times 10^{-1}$ | 89 | 11.33 |
| | 37 | DTGV | 3.25 | $3.7573 \times 10^{-2}$ | 7.4431 | $8.5823 \times 10^{-1}$ | 122 | 15.45 |
| | | TGV | 22.5 | $4.1719 \times 10^{-2}$ | 6.5339 | $8.4061 \times 10^{-1}$ | 52 | 6.64 |
| Gaussian | 43 | DTGV | 25 | $1.5530 \times 10^{-2}$ | 9.1966 | $9.7829 \times 10^{-1}$ | 56 | 7.17 |
| | | TGV | 100 | $1.8100 \times 10^{-2}$ | 7.8667 | $9.7200 \times 10^{-1}$ | 45 | 5.76 |
| | 37 | DTGV | 3 | $2.5498 \times 10^{-2}$ | 9.0841 | $9.2994 \times 10^{-1}$ | 90 | 11.41 |
| | | TGV | 17.5 | $3.0674 \times 10^{-2}$ | 7.4788 | $9.0199 \times 10^{-1}$ | 53 | 6.76 |
| `grass` | | | | | | | | |
| Out-of-focus | 43 | DTGV | 60 | $3.6313 \times 10^{-2}$ | 7.7364 | $8.7262 \times 10^{-1}$ | 136 | 15.55 |
| | | TGV | 550 | $3.6575 \times 10^{-2}$ | 7.6738 | $8.7188 \times 10^{-1}$ | 179 | 20.39 |
| | 37 | DTGV | 50 | $5.6164 \times 10^{-2}$ | 4.7390 | $7.6165 \times 10^{-1}$ | 160 | 18.56 |
| | | TGV | 55 | $5.7604 \times 10^{-2}$ | 4.5191 | $7.4566 \times 10^{-1}$ | 72 | 8.31 |
| Gaussian | 43 | DTGV | 65 | $2.9883 \times 10^{-2}$ | 6.3343 | $9.2764 \times 10^{-1}$ | 106 | 12.08 |
| | | TGV | 650 | $3.0814 \times 10^{-2}$ | 6.0676 | $9.2523 \times 10^{-1}$ | 136 | 15.48 |
| | 37 | DTGV | 5.5 | $4.2274 \times 10^{-2}$ | 4.7973 | $8.5615 \times 10^{-1}$ | 98 | 11.13 |
| | | TGV | 35 | $4.3936 \times 10^{-2}$ | 4.4624 | $8.4795 \times 10^{-1}$ | 54 | 6.18 |
| `leaves` | | | | | | | | |
| Out-of-focus | 43 | DTGV | 125 | $6.2767 \times 10^{-2}$ | 7.4978 | $8.2099 \times 10^{-1}$ | 251 | 31.18 |
| | | TGV | 1100 | $8.2397 \times 10^{-2}$ | 5.1342 | $7.1557 \times 10^{-1}$ | 435 | 53.74 |
| | 37 | DTGV | 12.5 | $9.5597 \times 10^{-2}$ | 4.1497 | $6.3065 \times 10^{-1}$ | 257 | 31.87 |
| | | TGV | 90 | $1.1874 \times 10^{-1}$ | 2.2665 | $4.3294 \times 10^{-1}$ | 113 | 14.03 |
| Gaussian | 43 | DTGV | 150 | $7.3332 \times 10^{-2}$ | 4.8675 | $7.7456 \times 10^{-1}$ | 236 | 29.13 |
| | | TGV | 1750 | $8.0857 \times 10^{-2}$ | 4.0190 | $7.3001 \times 10^{-1}$ | 380 | 46.77 |
| | 37 | DTGV | 12.5 | $9.0999 \times 10^{-2}$ | 3.3907 | $6.6469 \times 10^{-1}$ | 148 | 18.36 |
| | | TGV | 100 | $1.0308 \times 10^{-1}$ | 2.3081 | $5.6534 \times 10^{-1}$ | 103 | 12.85 |
| `carbon` | | | | | | | | |
| Out-of-focus | 43 | DTGV | 150 | $1.8360 \times 10^{-2}$ | $1.2830 \times 10^{1}$ | $9.4734 \times 10^{-1}$ | 331 | 13.78 |
| | | TGV | 850 | $2.3825 \times 10^{-2}$ | $1.0567 \times 10^{1}$ | $9.3671 \times 10^{-1}$ | 233 | 9.73 |
| | 37 | DTGV | 20 | $3.1682 \times 10^{-2}$ | 8.2416 | $8.6294 \times 10^{-1}$ | 171 | 7.07 |
| | | TGV | 150 | $3.8840 \times 10^{-2}$ | 6.4723 | $8.2237 \times 10^{-1}$ | 155 | 6.55 |
| Gaussian | 43 | DTGV | 250 | $2.0453 \times 10^{-2}$ | 8.6178 | $9.5974 \times 10^{-1}$ | 305 | 12.53 |
| | | TGV | 950 | $2.4839 \times 10^{-2}$ | 6.9302 | $9.5698 \times 10^{-1}$ | 171 | 7.12 |
| | 37 | DTGV | 15 | $2.7995 \times 10^{-2}$ | 6.2017 | $9.3007 \times 10^{-1}$ | 128 | 5.36 |
| | | TGV | 150 | $3.3061 \times 10^{-2}$ | 4.7572 | $8.9690 \times 10^{-1}$ | 118 | 4.73 |

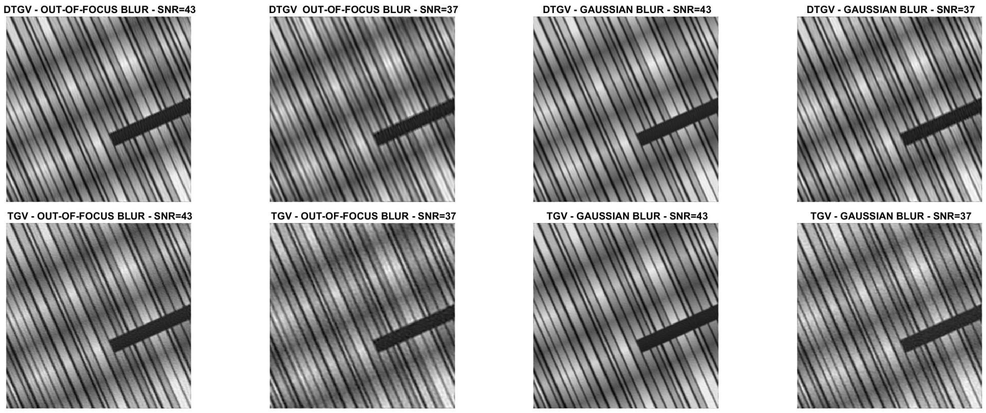

DTGV - OUT-OF-FOCUS BLUR - SNR=43
DTGV OUT-OF-FOCUS BLUR - SNR=37
DTGV - GAUSSIAN BLUR - SNR=43
DTGV - GAUSSIAN BLUR - SNR=37
TGV - OUT-OF-FOCUS BLUR - SNR=43
TGV - OUT-OF-FOCUS BLUR - SNR=37
TGV - GAUSSIAN BLUR - SNR=43
TGV - GAUSSIAN BLUR - SNR=37

**Figure 8.** Test problem `phantom`: images restored with DTGV$^2$ (**top**) and TGV$^2$ (**bottom**).

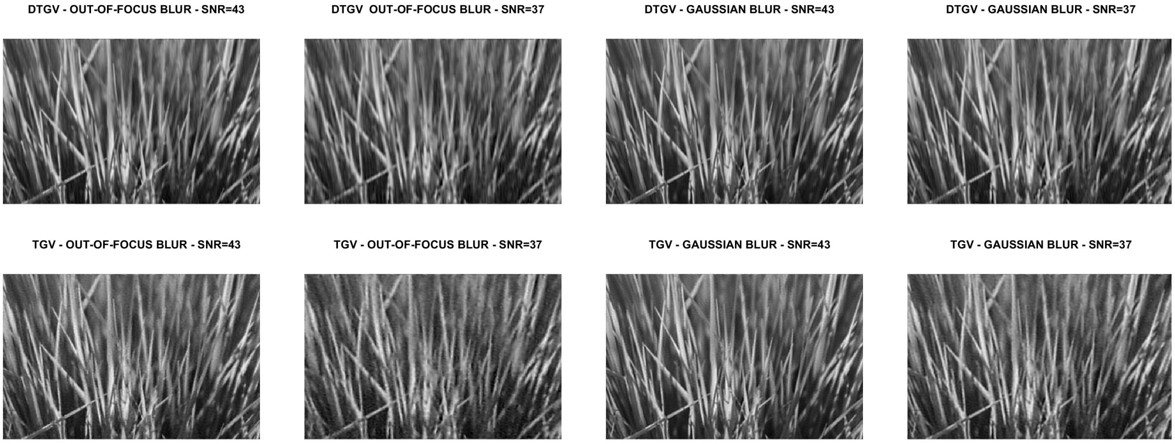

DTGV - OUT-OF-FOCUS BLUR - SNR=43
DTGV OUT-OF-FOCUS BLUR - SNR=37
DTGV - GAUSSIAN BLUR - SNR=43
DTGV - GAUSSIAN BLUR - SNR=37
TGV - OUT-OF-FOCUS BLUR - SNR=43
TGV - OUT-OF-FOCUS BLUR - SNR=37
TGV - GAUSSIAN BLUR - SNR=43
TGV - GAUSSIAN BLUR - SNR=37

**Figure 9.** Test problem `grass`: images restored with DTGV$^2$ (**top**) and TGV$^2$ (**bottom**).

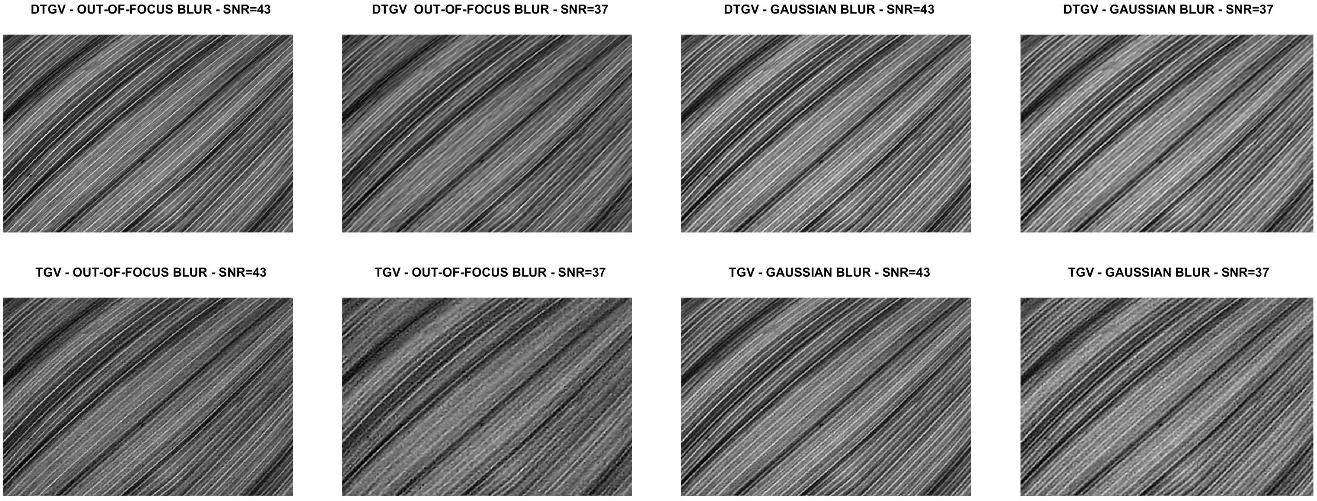

DTGV - OUT-OF-FOCUS BLUR - SNR=43
DTGV OUT-OF-FOCUS BLUR - SNR=37
DTGV - GAUSSIAN BLUR - SNR=43
DTGV - GAUSSIAN BLUR - SNR=37
TGV - OUT-OF-FOCUS BLUR - SNR=43
TGV - OUT-OF-FOCUS BLUR - SNR=37
TGV - GAUSSIAN BLUR - SNR=43
TGV - GAUSSIAN BLUR - SNR=37

**Figure 10.** Test problem `leaves`: images restored with DTGV$^2$ (**top**) and TGV$^2$ (**bottom**).

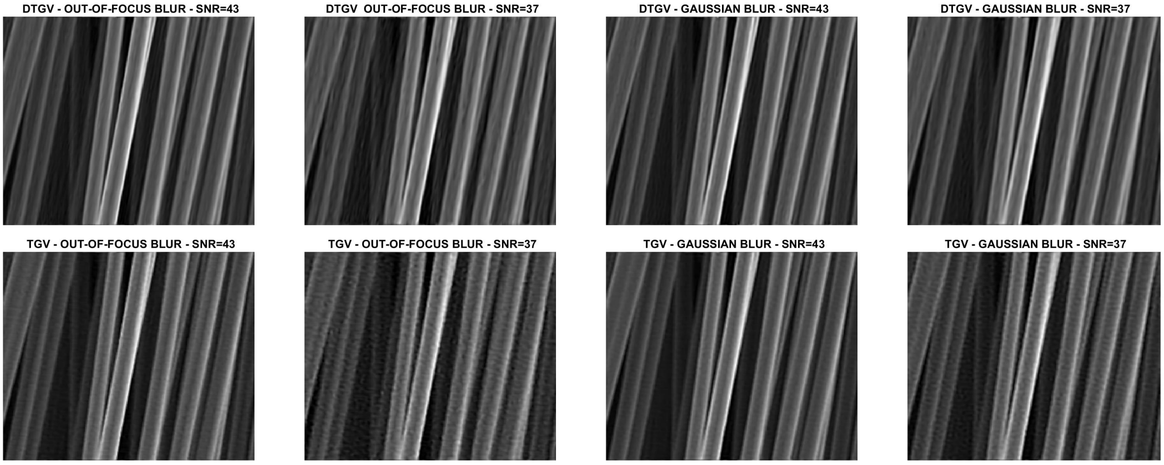

DTGV - OUT-OF-FOCUS BLUR - SNR=43
DTGV OUT-OF-FOCUS BLUR - SNR=37
DTGV - GAUSSIAN BLUR - SNR=43
DTGV - GAUSSIAN BLUR - SNR=37
TGV - OUT-OF-FOCUS BLUR - SNR=43
TGV - OUT-OF-FOCUS BLUR - SNR=37
TGV - GAUSSIAN BLUR - SNR=43
TGV - GAUSSIAN BLUR - SNR=37

**Figure 11.** Test problem `carbon`: images restored with DTGV$^2$ (**top**) and TGV$^2$ (**bottom**).

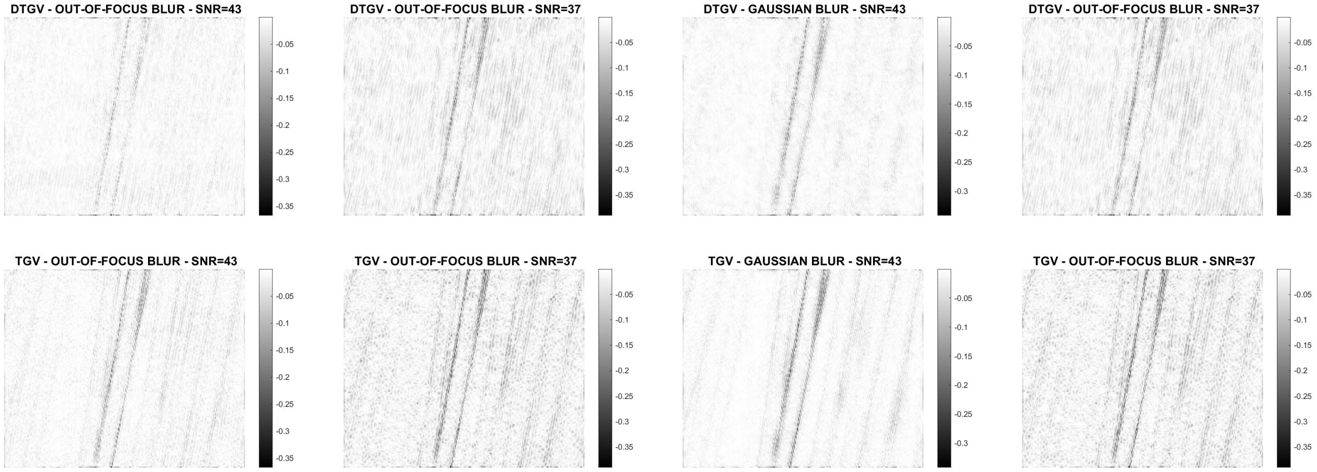

DTGV - OUT-OF-FOCUS BLUR - SNR=43
DTGV - OUT-OF-FOCUS BLUR - SNR=37
DTGV - GAUSSIAN BLUR - SNR=43
DTGV - OUT-OF-FOCUS BLUR - SNR=37
TGV - OUT-OF-FOCUS BLUR - SNR=43
TGV - OUT-OF-FOCUS BLUR - SNR=37
TGV - GAUSSIAN BLUR - SNR=43
TGV - OUT-OF-FOCUS BLUR - SNR=37
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35

**Figure 12.** Test problem `carbon`: difference images with DTGV$^2$ (**top**) and TGV$^2$ (**bottom**).

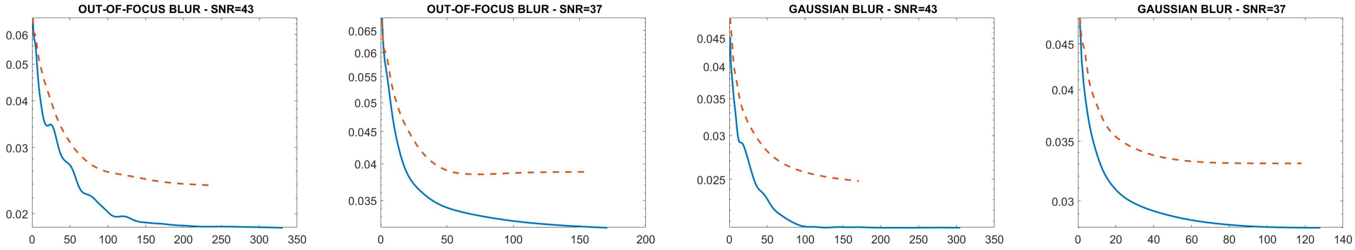


**Figure 13.** Test problem `carbon`: RMSE history for the KL-DGTV$^2$ (continuous line) and KL-TGV$^2$ (dashed line) models.

## 6. Conclusions

We dealt with the use of the Directional TGV regularization in the case of directional images corrupted by Poisson noise. We presented the KL-DTGV$^2$ model and introduced a two-block ADMM version for its minimization. Finally, we proposed an effective strategy for the estimation of the main direction of the image. Our numerical experiments show that for Poisson noise the DTGV$^2$ regularization provides superior restoration performance compared with the standard TGV$^2$ regularization, thus remarking the importance of taking into account the texture structure of the image. A crucial ingredient for the success of the model was the proposed direction estimation strategy, which proved to be more reliable than those proposed in the literature.

Possible future work includes the use of space-variant regularization terms and the analysis of automatic strategies for the selection of the regularization parameters.

**Author Contributions:** All authors have contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially supported by the Istituto Nazionale di Alta Matematica, Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS). D. di Serafino and M. Viola were also supported by the V:ALERE Program of the University of Campania "L. Vanvitelli".

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bertero, M.; Boccacci, P.; Desiderà, G.; Vicidomini, G. Image deblurring with Poisson data: From cells to galaxies. *Inverse Probl.* **2009**, *25*, 123006. [CrossRef]
2. Shepp, L.A.; Vardi, Y. Maximum Likelihood Reconstruction for Emission Tomography. *IEEE Trans. Med. Imaging* **1982**, *1*, 113–122. [CrossRef]
3. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physical D* **1992**, *60*, 259–268. [CrossRef]
4. Bredies, K.; Kunisch, K.; Pock, T. Total generalized variation. *SIAM J. Imaging Sci.* **2010**, *3*, 492–526. [CrossRef]
5. Bredies, K.; Holler, M. Regularization of linear inverse problems with total generalized variation. *J. Inverse Ill-Posed Probl.* **2014**, *22*, 871–913. [CrossRef]
6. Gao, Y.; Liu, F.; Yang, X. Total generalized variation restoration with non-quadratic fidelity. *Multidimens. Syst. Signal Process.* **2018**, *29*, 1459–1484. [CrossRef]
7. Di Serafino, D.; Landi, G.; Viola, M. TGV-based restoration of Poissonian images with automatic estimation of the regularization parameter. In Proceedings of the 21th International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 5–8 July 2021.
8. Bayram, I.; Kamasak, M.E. Directional Total Variation. *IEEE Signal Process. Lett.* **2012**, *19*, 781–784. [CrossRef]
9. Kongskov, R.D.; Dong, Y. Directional Total Generalized Variation Regularization for Impulse Noise Removal. In *Scale Space and Variational Methods in Computer Vision*; Lauze, F., Dong, Y., Dahl, A., Eds.; SSVM2017; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10302, pp. 221–231. [CrossRef]
10. Kongskov, R.D.; Dong, Y.; Knudsen, K. Directional total generalized variation regularization. *BIT* **2019**, *59*, 903–928. [CrossRef]
11. di Serafino, D.; Landi, G.; Viola, M. ACQUIRE: An inexact iteratively reweighted norm approach for TV-based Poisson image restoration. *Appl. Math. Comput.* **2020**, *364*, 124678. [CrossRef]
12. Setzer, S.; Steidl, G.; Teuber, T. Restoration of images with rotated shapes. *Numer. Algorithms* **2008**, *48*, 49–66. [CrossRef]
13. Kanopoulos, N.; Vasanthavada, N.; Baker, R. Design of an image edge detection filter using the Sobel operator. *IEEE J. Solid-State Circuits* **1988**, *23*, 358–367. [CrossRef]
14. Hough, P. Method and Means for Recognizing Complex Patterns. U.S. Patent 3069654, 18 December 1962.
15. Duda, R.O.; Hart, P.E. Use of the Hough transform to detect lines and curves in pictures. *Commun. ACM* **1972**, *15*, 11–15. [CrossRef]
16. Bonettini, S.; Zanella, R.; Zanni, L. A scaled gradient projection method for constrained image deblurring. *Inverse Probl.* **2009**, *25*, 015002. [CrossRef]
17. Birgin, E.G.; Martínez, J.M.; Raydan, M. Spectral Projected Gradient Methods: Review and Perspectives. *J. Stat. Softw.* **2014**, *60*, 1–21. [CrossRef]
18. di Serafino, D.; Ruggiero, V.; Toraldo, G.; Zanni, L. On the steplength selection in gradient methods for unconstrained optimization. *Appl. Math. Comput.* **2018**, *318*, 176–195. [CrossRef]
19. Goldstein, T.; Osher, S. The Split Bregman Method for L1-Regularized Problems. *SIAM J. Imaging Sci.* **2009**, *2*, 323–343. [CrossRef]
20. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [CrossRef]
21. Parikh, N.; Boyd, S. Proximal Algorithms. *Found. Trends Optim.* **2014**, *1*, 123–231. [CrossRef]
22. Bonettini, S.; Loris, I.; Porta, F.; Prato, M. Variable Metric Inexact Line-Search-Based Methods for Nonsmooth Optimization. *SIAM J. Optim.* **2016**, *26*, 891–921. [CrossRef]
23. De Simone, V.; di Serafino, D.; Viola, M. A subspace-accelerated split Bregman method for sparse data recovery with joint $\ell_1$-type regularizers. *Electron. Trans. Numer. Anal.* **2020**, *53*, 406–425. [CrossRef]

24. Setzer, S.; Steidl, G.; Teuber, T. Deblurring Poissonian images by split Bregman techniques. *J. Vis. Commun. Image Represent.* **2010**, *21*, 193–199. [CrossRef]
25. Eckstein, J.; Bertsekas, D.P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **1992**, *55*, 293–318. [CrossRef]
26. Beck, A. *First-Order Methods in Optimization*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2017; [CrossRef]
27. Nagy, J.G.; Palmer, K.; Perrone, L. Iterative methods for image deblurring: A Matlab object-oriented approach. *Numer. Algorithms* **2004**, *36*, 73–93. [CrossRef]
28. Bertero, M.; Boccacci, P.; Ruggiero, V. *Inverse Imaging with Poisson Data*; IOP Publishing: Bristol, UK, 2018; pp. 2053–2563. [CrossRef]
29. Banham, M.; Katsaggelos, A. Digital image restoration. *IEEE Signal Process. Mag.* **1997**, *14*, 24–41. [CrossRef]
30. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]