

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

di Serafino, D., Landi, G., Viola, M. (2020). ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration. APPLIED MATHEMATICS AND COMPUTATION, 364, 1-23 [10.1016/j.amc.2019.124678].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/698078> since: 2020-01-08

*Published:*

DOI: <http://doi.org/10.1016/j.amc.2019.124678>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

**Daniela di Serafino, Germana Landi, Marco Viola, ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration, Applied Mathematics and Computation, Volume 364, 2020, 124678, ISSN 0096-3003.**

The final published version is available online at:  
<https://doi.org/10.1016/j.amc.2019.124678>

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

*This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)*

***When citing, please refer to the published version.***

# ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration<sup>\*</sup>

Daniela di Serafino<sup>a</sup>, Germana Landi<sup>b</sup>, Marco Viola<sup>c</sup>

<sup>a</sup>*Dipartimento di Matematica e Fisica, Università degli Studi della Campania  
"Luigi Vanvitelli", viale A. Lincoln 5, 81100 Caserta, Italy*

<sup>b</sup>*Dipartimento di Matematica, Università degli Studi di Bologna, Piazza di Porta  
S. Donato 5, 40126 Bologna, Italy*

<sup>c</sup>*Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti",  
Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy*

---

## Abstract

We propose a method, called ACQUIRE, for the solution of constrained optimization problems modeling the restoration of images corrupted by Poisson noise. The objective function is the sum of a generalized Kullback-Leibler divergence term and a TV regularizer, subject to nonnegativity and possibly other constraints, such as flux conservation. ACQUIRE is a line-search method that considers a smoothed version of TV, based on a Huber-like function, and computes the search directions by minimizing quadratic approximations of the problem, built by exploiting some second-order information. A classical second-order Taylor approximation is used for the Kullback-Leibler term and an iteratively reweighted norm approach for the smoothed TV term. We prove that the sequence generated by the method has a subsequence converging to a minimizer of the smoothed problem and any limit point is a minimizer. Furthermore, if the problem is strictly convex, the whole sequence is convergent. We note that convergence is achieved without requiring the exact minimization of the quadratic subproblems; low accuracy in this minimization can be used in practice, as shown by numerical results. Experiments on reference test problems show that our method is competitive with well-established methods for TV-based Poisson image restoration, in terms of both computational efficiency and image quality.

*Keywords:* image restoration, Poisson noise, TV regularization, iteratively reweighted norm approach, quadratic approximation.

---

<sup>\*</sup>This work was partially supported by Gruppo Nazionale per il Calcolo Scientifico - Istituto Nazionale di Alta Matematica (GNCS-INdAM).

*Email addresses:* [daniela.diserafino@unicampania.it](mailto:daniela.diserafino@unicampania.it) (Daniela di Serafino),  
[germana.landi@unibo.it](mailto:germana.landi@unibo.it) (Germana Landi), [marco.viola@uniroma1.it](mailto:marco.viola@uniroma1.it) (Marco Viola)

## 1. Introduction

Restoring images corrupted by Poisson noise is required in many applications, such as fluorescence microscopy [47], X-ray computed tomography (CT) [33], positron emission tomography (PET) [51], confocal microscopy [44] and astronomical imaging [52, 4]. Thus, this is a very active research area in image processing. We consider a discrete formulation of the problem, where the object to be restored is represented by a vector  $\mathbf{x} \in \mathbb{R}^n$  and the measured data are assumed to be a vector  $\mathbf{y} \in \mathbb{N}_0^m$ , whose entries  $y_j$  are samples from  $m$  independent Poisson random variables  $Y_j$  with probability

$$P(Y_j = y_j) = \frac{e^{-(A\mathbf{x} + \mathbf{b})_j} (A\mathbf{x} + \mathbf{b})_j^{y_j}}{y_j!},$$

where the matrix  $A = (a_{ij}) \in \mathbb{R}^{m \times n}$  models the observation mechanism of the imaging system and  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{b} > 0$ , models the background radiation detected by the sensors. Standard assumptions on  $A$  are

$$a_{ij} \geq 0 \text{ for all } i, j, \quad \sum_{i=1}^m a_{ij} = 1 \text{ for all } j. \quad (1)$$

By applying a maximum-likelihood approach [4, 51], we can estimate  $\mathbf{x}$  by minimizing the Kullback-Leibler (KL) divergence of  $A\mathbf{x} + \mathbf{b}$  from  $\mathbf{y}$ :

$$D_{KL}(A\mathbf{x} + \mathbf{b}, \mathbf{y}) = \sum_{j=1}^m \left( y_j \ln \frac{y_j}{(A\mathbf{x} + \mathbf{b})_j} + (A\mathbf{x} + \mathbf{b})_j - y_j \right), \quad (2)$$

where we set  $y_j \ln(y_j/(A\mathbf{x} + \mathbf{b})_j) = 0$  if  $y_j = 0$ . A regularization term is usually added to (2) to deal with the inherent ill-conditioning of the estimation problem. We focus on edge-preserving regularization by Total Variation (TV) [46], which has received considerable attention because of its ability of preserving edges and smoothing flat areas of the images. We note that, although TV regularization is known to suffer from undesirable staircase artifacts, it is still widely used in many medical and biological applications (see, e.g., [1, 40, 60], [http://ranger.uta.edu/~huang/R\\_CSMRI.htm](http://ranger.uta.edu/~huang/R_CSMRI.htm)). Furthermore, by focusing on TV-regularized problems, we introduce and test a novel solution method that allows for extensions to other models, such as high-order TV [36, 43] and Total Generalized Variation [12, 13], proposed to reduce the staircase effect.

Assuming, for simplicity, that  $\mathbf{x}$  is obtained by stacking the columns of a 2D image  $X = (X_{k,l}) \in \mathbb{R}^{r \times s}$ , i.e.,  $x_i = X_{k,l}$  with  $i = (l-1)r + k$  and  $n = rs$ , the following discrete version of the TV functional can be defined [16]:

$$TV(X) = \sum_{k=1}^r \sum_{l=1}^s \sqrt{(X_{k+1,l} - X_{k,l})^2 + (X_{k,l+1} - X_{k,l})^2},$$

where  $X$  is supposed to satisfy some boundary conditions, e.g., periodic. This

can be also written as

$$TV(\mathbf{x}) = \sum_{i=1}^n \|D_i \mathbf{x}\|, \quad (3)$$

where

$$D_i = \begin{pmatrix} \mathbf{e}_{(l-1)r+k+1}^T - \mathbf{e}_{(l-1)r+k}^T \\ \mathbf{e}_{lr+k}^T - \mathbf{e}_{(l-1)r+k}^T \end{pmatrix}, \quad i = (l-1)r + k,$$

$\mathbf{e}_q \in \mathbb{R}^n$  is the  $q$ th standard basis vector, and  $\|\cdot\|$  is the 2-norm

Thus, we are interested in solving the following problem:

$$\begin{aligned} & \text{minimize} && D_{KL}(\mathbf{x}) + \lambda TV(\mathbf{x}), \\ & \text{s.t.} && \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (4)$$

where  $D_{KL}(\mathbf{x})$  is a shorthand for  $D_{KL}(A\mathbf{x} + \mathbf{b}, \mathbf{y})$ ,  $\lambda > 0$  is a regularization parameter, and  $\mathbf{x} \in \mathcal{S}$  corresponds to some physical constraints. The nonnegativity of the image intensity naturally leads to the constraint  $\mathbf{x} \geq \mathbf{0}$ . When the matrix  $A$  comes from the discretization of a convolution operator and it is normalized as in (1), the constraint  $\sum_{i=1}^n x_i = \sum_{j=1}^m (y_j - b_j)$  can be added, since the convolution performs a modification of the intensity distribution, while the total intensity remains constants [6].<sup>1</sup> In other words, common choices of  $\mathcal{S}$  are

$$\mathcal{S} = \mathcal{S}_1 := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}\} \quad (5)$$

or

$$\mathcal{S} = \mathcal{S}_2 := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}, \mathbf{e}^T \mathbf{x} = \bar{\mathbf{e}}^T (\mathbf{y} - \mathbf{b})\}, \quad (6)$$

where  $\mathbf{e}$  and  $\bar{\mathbf{e}}$  denote the vectors of all 1's of sizes  $n$  and  $m$ , respectively.

Various approaches have been proposed to solve problem (4), mostly with  $\mathcal{S} = \mathcal{S}_1$ ; a key issue in all cases is to deal with the nondifferentiability of the TV functional. Some representative methods are listed next. A classical approach consists in approximating the TV functional with a smooth version of it and using well-established techniques such as expectation-maximization methods [34, 42], gradient-projection methods with suitable scaling techniques aimed at accelerating convergence [11, 37, 58], and alternating linearized minimization methods [35].

The approximation of TV can be avoided, e.g., by using forward-backward splitting techniques; this is the case of the proximal-gradient methods proposed in [9, 32] and the forward-backward EM method discussed in [48]. On the other hand, the previous methods require, at each step, the solution of a Rudin-Osher-Fatemi (ROF) denoising subproblem [46], which can be computed only approximately, using, e.g., the algorithms proposed in [3, 16]. Methods based on ADMM and SPLIT BREGMAN techniques, such as those presented in [26, 29, 49], do not exploit smooth TV approximations too. They generally use

---

<sup>1</sup>We have implicitly assumed that  $\mathbf{y}$  has been converted into a real vector with entries ranging in the same interval as the entries of  $\mathbf{x}$ .

more memory because of auxiliary variables of the same size as  $\mathbf{x}$  or  $\mathbf{y}$ , and require the solution of linear systems involving  $A^T A$  and, possibly, the solution of ROF subproblems. Finally, a different approach to avoid the difficulties associated with the nondifferentiability of the TV functional is based on the idea of reformulating (4) as a saddle-point problem and solving it by a primal-dual algorithm. In this context, an alternating extragradient scheme has been presented in [10], and a procedure exploiting the Chambolle-Pock algorithm [17] has been described in [56].

In this paper we take a different approach, aimed at exploiting some second-order information not considered by the aforementioned methods. We consider a smoothed version of TV, based on a Huber-like function, and propose a line-search method, called ACQUIRE (Algorithm based on Consecutive QUadratic and Iteratively REweighted norm approximations), which minimizes a sequence of quadratic models obtained by a second-order Taylor approximation of the KL divergence and an iteratively reweighted norm (IRN) approximation of the smoothed TV. We prove the convergence of ACQUIRE with inexact solution of the inner quadratic problems. We show by numerical experiments that exploiting some second-order information can lead to fast image restorations even with low accuracy requirements on the solution of the inner problems, without affecting the quality of the reconstructed images. In particular, ACQUIRE generally produces a strong reduction of the reconstruction error in the first iterations, thus achieving a good tradeoff between accuracy and efficiency, and resulting competitive with state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2 we recall some preliminary concepts that will be exploited later. In Section 3 we describe our method and in Section 4 we prove that it is well posed and convergent. We provide implementation details and discuss the results obtained by applying the proposed method to several test problems in Section 5. Some conclusions are reported in Section 6.

## 2. Preliminaries

We first provide some useful details about the KL divergence and introduce a smooth version of the TV functional. Then we recall the concept of projected gradient and its basic properties, exploited later in this work.

Assumptions (1) and  $\mathbf{b} > 0$  ensure that, for any given  $\mathbf{y} \geq 0$ ,  $D_{KL}$  is a nonnegative, convex, coercive, twice continuously differentiable function in  $\mathbb{R}_+^n$  (see, e.g., [5, 26]). Its gradient and Hessian are given by

$$\nabla D_{KL}(\mathbf{x}) = A^T \left( \mathbf{e} - \frac{\mathbf{y}}{A\mathbf{x} + \mathbf{b}} \right)$$

and

$$\nabla^2 D_{KL}(\mathbf{x}) = A^T U(\mathbf{x})^2 A, \quad U(\mathbf{x}) = \text{diag} \left( \frac{\sqrt{\mathbf{y}}}{A\mathbf{x} + \mathbf{b}} \right), \quad (7)$$

where the square root and the ratios are intended componentwise, and  $\text{diag}(\mathbf{v})$  denotes the diagonal matrix with diagonal entries equal to the entries of  $\mathbf{v}$ . It can be proved that  $\nabla D_{KL}$  is Lipschitz continuous [32]; furthermore, it follows from (7) that  $\nabla^2 D_{KL}$  is positive definite, i.e.,  $D_{KL}$  is strictly convex, whenever  $\mathbf{y} > \mathbf{0}$  and  $A$  has nullspace  $\mathcal{N}(A) = \{\mathbf{0}\}$ . In this case, if  $\mathbf{x}$  is constrained to be in a bounded subset of the nonnegative orthant, e.g., the set  $\mathcal{S}_2$  in (6), the minimum eigenvalue of  $\nabla^2 D_{KL}(\mathbf{x})$  is bounded below independently of  $\mathbf{x}$ , and  $D_{KL}$  is strongly convex.

From a practical point of view, it is interesting to note that  $A$  is usually the representation of a convolution operator, and hence the computation of  $\nabla D_{KL}$  or of matrix-vector products involving  $\nabla^2 D_{KL}$  can be performed efficiently via fast algorithms for discrete Fourier, cosine or sine transforms.

The  $TV$  functional is nonnegative, convex and continuous. Thus problem (4) admits a solution, which is unique if  $\mathbf{y} > \mathbf{0}$  and  $\mathcal{N}(A) = \{\mathbf{0}\}$ . Since  $TV$  is not differentiable, we use a regularized version of it,  $TV_\mu$ . Taking into account the discussion in [55] about smoothed versions of  $TV$ , we consider

$$TV_\mu(\mathbf{x}) = \sum_{i=1}^n \phi_\mu(\|D_i \mathbf{x}\|),$$

where  $\phi_\mu$  is the Huber-like function

$$\phi_\mu(z) = \begin{cases} |z| & \text{if } |z| > \mu, \\ \frac{1}{2}(\frac{z^2}{\mu} + \mu) & \text{otherwise,} \end{cases}$$

and  $\mu > 0$  is small. It is easy to verify that  $TV_\mu$  is Lipschitz continuously differentiable and its gradient reads as follows:

$$\nabla TV_\mu(\mathbf{x}) = \sum_{i=1}^n \nabla \phi_\mu(\|D_i \mathbf{x}\|), \quad \nabla \phi_\mu(\|D_i \mathbf{x}\|) = \begin{cases} \frac{D_i^T D_i \mathbf{x}}{\|D_i \mathbf{x}\|} & \text{if } \|D_i \mathbf{x}\| > \mu, \\ \frac{D_i^T D_i \mathbf{x}}{\mu} & \text{otherwise.} \end{cases}$$

We also observe that  $TV_\mu$  is not twice continuously differentiable, but has continuous Hessian for all  $\mathbf{x}$  such that  $\|D_i \mathbf{x}\| \neq \mu$ :

$$\begin{aligned} \nabla^2 TV_\mu(\mathbf{x}) &= \sum_{i=1}^n \nabla^2 \phi_\mu(\|D_i \mathbf{x}\|), \\ \nabla^2 \phi_\mu(\|D_i \mathbf{x}\|) &= \begin{cases} \frac{D_i^T D_i}{\|D_i \mathbf{x}\|} - \frac{(D_i^T D_i \mathbf{x})(D_i^T D_i \mathbf{x})^T}{\|D_i \mathbf{x}\|^3} & \text{if } \|D_i \mathbf{x}\| > \mu, \\ \frac{D_i^T D_i}{\mu} & \text{if } \|D_i \mathbf{x}\| < \mu. \end{cases} \end{aligned} \quad (8)$$

Now we recall basic notions about the projected gradient. Let  $\mathcal{S}$  be a

nonempty, closed and convex set. For any continuously differentiable function  $f : \mathcal{D} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ , with  $\mathcal{D}$  open set containing  $\mathcal{S}$ , the projected gradient of  $f$  at  $\mathbf{x} \in \mathcal{S}$  is defined as the orthogonal projection of  $-\nabla f$  onto the tangent cone to  $\mathcal{S}$  at  $\mathbf{x}$ , denoted by  $T_{\mathcal{S}}(\mathbf{x})$ :

$$\nabla_{\mathcal{S}} f(\mathbf{x}) = \arg \min \{ \|\mathbf{v} + \nabla f(\mathbf{x})\| \mid \mathbf{v} \in T_{\mathcal{S}}(\mathbf{x}) \},$$

When  $\mathcal{S}$  is the set  $\mathcal{S}_1$  defined in (5), the tangent cone takes the form

$$T_{\mathcal{S}}(\mathbf{x}) = \{ \mathbf{v} \in \mathbb{R}^n : v_i \geq 0 \text{ if } x_i = 0 \}$$

and the computation of  $\nabla_{\mathcal{S}} f(\mathbf{x})$  is straightforward; when  $\mathcal{S}$  is the set  $\mathcal{S}_2$  in (6),

$$T_{\mathcal{S}}(\mathbf{x}) = \{ \mathbf{v} \in \mathbb{R}^n : \mathbf{e}^T \mathbf{v} = 0 \text{ and } v_i \geq 0 \text{ if } x_i = 0 \},$$

and  $\nabla_{\mathcal{S}} f(\mathbf{x})$  can be efficiently determined too, thanks to the availability of low-cost algorithms for computing the projection in this case (see, e.g., [15, 18, 19]).

Since the projection onto  $\mathcal{S}$  is nonexpansive, for all  $\mathbf{x}, \bar{\mathbf{x}} \in \mathcal{S}$  it is

$$\|\nabla_{\mathcal{S}} f(\mathbf{x}) - \nabla_{\mathcal{S}} f(\bar{\mathbf{x}})\| \leq \|\mathbf{x} - \bar{\mathbf{x}}\|;$$

furthermore,

$$-\nabla f(\mathbf{x}) = \nabla_{\mathcal{S}} f(\mathbf{x}) + P_{N_{\mathcal{S}}(\mathbf{x})}(-\nabla f(\mathbf{x})), \quad (9)$$

where  $P_{N_{\mathcal{S}}(\mathbf{x})}$  denotes the orthogonal projection operator onto the normal cone to  $\mathcal{S}$  at  $\mathbf{x}$ ,

$$N_{\mathcal{S}}(\mathbf{x}) = \{ \mathbf{v} \in \mathbb{R}^n : \mathbf{v}^T \mathbf{u} \leq 0 \text{ for all } \mathbf{u} \in \mathcal{S}(\mathbf{x}) \},$$

which is the polar cone of  $T_{\mathcal{S}}(\mathbf{x})$  (see, e.g., [59, Lemma 2.2]).

Finally, it is well known that any constrained stationary point  $\mathbf{x}^*$  of  $f$  in  $\mathcal{S}$  is characterized by  $\nabla_{\mathcal{S}} f(\mathbf{x}^*) = \mathbf{0}$  and that  $\|\nabla_{\mathcal{S}} f\|$  is lower semicontinuous on  $\mathcal{S}$  (see, e.g., [14]).

### 3. IRN-based inexact minimization method

We propose an iterative method for solving the problem

$$\begin{aligned} & \text{minimize} && D_{KL}(\mathbf{x}) + \lambda TV_{\mu}(\mathbf{x}), \\ & \text{s.t.} && \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (10)$$

where  $\mathcal{S}$  can be any nonempty, closed and convex subset of  $\mathbb{R}_+^n$ , although our practical interest is for the feasible sets in (5)-(6). This method is based on two main steps: the inexact solution of a quadratic model of (4) and a line-search procedure.

Given an iterate  $\mathbf{x}^{(k)} \in \mathcal{S}$ , we consider the following quadratic approximation



of  $D_{KL}$ :

$$\begin{aligned} D_{KL}(\mathbf{x}) &\approx D_{KL}^{(k)}(\mathbf{x}) = D_{KL}(\mathbf{x}^{(k)}) + (\mathbf{x} - \mathbf{x}^{(k)})^T \nabla D_{KL}(\mathbf{x}^{(k)}) \\ &\quad + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^T (\nabla^2 D_{KL}(\mathbf{x}^{(k)}) + \gamma I)(\mathbf{x} - \mathbf{x}^{(k)}), \end{aligned} \quad (11)$$

where  $I$  is the identity matrix and  $\gamma > 0$ . Note that  $\gamma I$  has been introduced to ensure that  $D_{KL}^{(k)}$  is strongly convex; obviously, we can set  $\gamma = 0$  if  $\mathbf{y} > \mathbf{0}$ ,  $\mathcal{N}(A) = \{\mathbf{0}\}$  and  $\mathcal{S}$  is bounded.

In order to build a quadratic model of  $TV_\mu$ , we use the IRN approach described in [45], i.e., we approximate  $TV_\mu(\mathbf{x})$  as follows:

$$TV_\mu(\mathbf{x}) \approx TV_\mu^{(k)}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n w_i^{(k)} \|D_i \mathbf{x}\|^2 + \frac{1}{2} TV_\mu(\mathbf{x}^{(k)}),$$

where

$$w_i^{(k)} = \begin{cases} \|D_i \mathbf{x}^{(k)}\|^{-1} & \text{if } \|D_i \mathbf{x}^{(k)}\| > \mu, \\ \mu^{-1} & \text{otherwise.} \end{cases}$$

Trivially,

$$TV_\mu^{(k)}(\mathbf{x}^{(k)}) = TV_\mu(\mathbf{x}^{(k)}), \quad \nabla TV_\mu^{(k)}(\mathbf{x}^{(k)}) = \nabla TV_\mu(\mathbf{x}^{(k)});$$

furthermore,

$$\nabla^2 TV_\mu^{(k)}(\mathbf{x}^{(k)}) = \sum_{i=1}^n w_i^{(k)} D_i^T D_i,$$

and hence, for any  $\mathbf{x}$  such that  $\|D_i \mathbf{x}^{(k)}\| \neq \mu$ , the Hessian  $\nabla^2 TV_\mu^{(k)}(\mathbf{x}^{(k)})$  can be regarded as an approximation of  $\nabla^2 TV_\mu(\mathbf{x}^{(k)})$ , obtained by neglecting the higher order term in the right-hand side of (8), which generally increases the ill-conditioning of the Hessian matrix. Thus, we can say that  $TV_\mu^{(k)}$  contains some second-order information about  $TV_\mu$ . It is worth noting that the higher order term of the Hessian of a smoothed TV function is also neglected in the lagged diffusivity method by Vogel and Oman [53].

In the following, to simplify the notation we set

$$\begin{aligned} F(\mathbf{x}) &= D_{KL}(\mathbf{x}) + \lambda TV_\mu(\mathbf{x}), \\ F_k(\mathbf{x}) &= D_{KL}^{(k)}(\mathbf{x}) + \lambda TV_\mu^{(k)}(\mathbf{x}). \end{aligned}$$

At iteration  $k$ , our method computes a feasible approximation  $\hat{\mathbf{x}}^{(k)}$  to the solution  $\bar{\mathbf{x}}^{(k)}$  of the quadratic problem

$$\begin{aligned} &\text{minimize} && F_k(\mathbf{x}), \\ &\text{s.t.} && \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (12)$$

and performs a line search along the direction

$$\mathbf{d}^{(k)} = \widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)},$$

until an Armijo condition is satisfied, to obtain an approximation  $\mathbf{x}^{(k+1)}$  to the solution of problem (10). This procedure is sketched in Algorithm 1 and is called ACQUIRE, which comes from “Algorithm based on Consecutive QUadratic and Iteratively REweighted norm approximations”.

---

**Algorithm 1** – ACQUIRE

---

```

1: choose  $\mathbf{x}_0 \in \mathcal{S}$ ,  $\eta \in (0, 1)$ ,  $\delta \in (0, 1)$ ,  $\{\varepsilon_k\}$  such that  $\varepsilon_k > 0$  and  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ 
2: for  $k = 1, 2, \dots$  do
3:   compute an approximate solution  $\widehat{\mathbf{x}}^{(k)} \in \mathcal{S}$  to the quadratic problem (12),
      such that
4:      $\alpha_k := 1$ 
5:      $\mathbf{d}^{(k)} := \widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}$ 
6:      $\mathbf{x}_\alpha^{(k)} := \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ 
7:     while  $F(\mathbf{x}_\alpha^{(k)}) > F(\mathbf{x}^{(k)}) + \eta \alpha_k \nabla F(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)}$  do
8:        $\alpha_k := \delta \alpha_k$ 
9:        $\mathbf{x}_\alpha^{(k)} := \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ 
10:    end while
11:     $\mathbf{x}^{(k+1)} = \mathbf{x}_\alpha^{(k)}$ 
12: end for
```

---

ACQUIRE is well posed (i.e., a steplength  $\alpha_k$  satisfying the Armijo condition can be found in a finite number of iterations) and is convergent; this is proved in Section 4. Step 3 does not require the exact solution of problem (12), but only the computation of an approximate solution such that condition (13) at line 3 of the algorithm holds, with  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ .

In Section 4 we also show that the first condition in (13) is satisfied if

$$\|\nabla_{\mathcal{S}} F_k(\widehat{\mathbf{x}}^{(k)})\| \leq \theta^k \|\nabla_{\mathcal{S}} F_k(\mathbf{x}^{(0)})\|, \quad (14)$$

and  $\theta \in (0, 1)$ . Therefore, the first condition in (13) can be replaced by another one which is simple to verify when the projected gradient can be easily computed, e.g., in the practical cases where  $\mathcal{S}$  is one of the sets in (5)-(6).

The second condition in (13) can be achieved by using any constrained minimization algorithm. We note that, for the restoration problems considered in this work, gradient-projection methods, such as those in [11, 24, 39], are suited to the solution of the inner problems (12). Indeed, numerical experiments have shown that very low accuracy is required in practice in the solution of the inner problems; furthermore, the computational cost per iteration of gradient pro-

jection methods is modest when low-cost algorithms for the projection onto the feasible set are available. More details on the inner method used in our experiments are given in Section 5.

#### 4. Well-posedness and convergence

In order to prove that ACQUIRE is well posed, we need the following lemma [7, Lemma A24].

**Lemma 1 (Descent lemma).** *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be continuously differentiable and let  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . If there exists  $L > 0$  such that*

$$\|\nabla f(\mathbf{x} + t\mathbf{y}) - \nabla f(\mathbf{x})\| \leq Lt\|\mathbf{y}\| \text{ for all } t \in [0, 1],$$

then

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{y} + \frac{L}{2} \|\mathbf{y}\|^2.$$

We also observe that, at step 3 of Algorithm 1, we can find  $\widehat{\mathbf{x}}^{(k)} \neq \mathbf{x}^{(k)}$  unless  $\mathbf{x}^{(k)}$  is the solution  $\bar{\mathbf{x}}^{(k)}$  of problem (12). However, in this case  $\mathbf{x}^{(k)}$  is the solution of problem (10), since the gradients, and hence the projected gradients, of the objective functions of the two problems coincide at  $\mathbf{x}^{(k)}$ . Therefore, in the following we can assume that  $\widehat{\mathbf{x}}^{(k)} \neq \mathbf{x}^{(k)}$ .

The next theorem shows that the steplength  $\alpha_k$  required to obtain the iterate  $\mathbf{x}^{(k+1)}$  can be found after a finite number of steps and that it is bounded away from zero.

**Theorem 2.** *Let  $\delta \in (0, 1)$ . There exist  $\bar{\alpha} > 0$  independent of  $k$  and an integer  $j_k \geq 0$  such that for  $\alpha_k = \delta^{j_k}$*

$$F(\mathbf{x}_{\alpha}^{(k)}) \leq F(\mathbf{x}^{(k)}) + \eta \alpha_k \nabla F(\mathbf{x}^{(k)})^T (\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}), \quad (15)$$

$$\alpha_k \geq \bar{\alpha}. \quad (16)$$

*Proof.* For  $F$  has Lipschitz continuous gradient, by applying Lemma 1 we get

$$F(\mathbf{x}_{\alpha}^{(k)}) \leq F(\mathbf{x}^{(k)}) + \alpha_k \nabla F(\mathbf{x}^{(k)})^T (\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) + \alpha_k^2 \frac{L}{2} \|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2,$$

where  $L$  is the Lipschitz constant of  $\nabla F$ . Then, (15) holds if we find  $\alpha_k$  such that

$$\nabla F(\mathbf{x}^{(k)})^T (\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) + \alpha_k \frac{L}{2} \|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2 \leq \eta \nabla F(\mathbf{x}^{(k)})^T (\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}),$$

or, equivalently,

$$(1 - \eta) \nabla F(\mathbf{x}^{(k)})^T (\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) + \alpha_k \frac{L}{2} \|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2 \leq 0. \quad (17)$$

From  $\nabla F_k(\mathbf{x}^{(k)}) = \nabla F(\mathbf{x}^{(k)})$ , the strong convexity of  $F_k$  and step 3 of Algorithm 1, it follows that

$$\begin{aligned}\nabla F(\mathbf{x}^{(k)})^T(\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) &= \nabla F_k(\mathbf{x}^{(k)})^T(\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) \\ &\leq F_k(\widehat{\mathbf{x}}^{(k)}) - F_k(\mathbf{x}^{(k)}) - \frac{\gamma}{2}\|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2 \quad (18) \\ &\leq -\frac{\gamma}{2}\|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2,\end{aligned}$$

where  $\gamma$  is the strong convexity parameter of  $F_k$ . Thus, (17) holds for any  $\alpha_k$  such that

$$\frac{\gamma}{2}(\eta - 1)\|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2 + \alpha_k \frac{L}{2}\|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2 \leq 0.$$

By choosing the first nonnegative integer  $j_k$  such that

$$\delta^{j_k} \leq \min \left\{ 1, \frac{\gamma(1-\eta)}{L} \right\}$$

and setting

$$\overline{\alpha} = \min \left\{ 1, \frac{\delta \gamma(1-\eta)}{L} \right\}$$

we get the thesis.  $\square$

Now we prove that the sequence generated by ACQUIRE has a subsequence converging to a solution of problem (10). Because of the convexity of  $F$ , it is sufficient to prove that the subsequence converges to a constrained stationary point of  $F$ .

**Theorem 3.** *Let  $\{\mathbf{x}^{(k)}\}$  be the sequence generated by Algorithm 1. Then there exists a subsequence  $\{\mathbf{x}^{(k_j)}\}$  such that*

$$\lim_{k_j \rightarrow \infty} \mathbf{x}^{(k_j)} = \overline{\mathbf{x}},$$

where  $\overline{\mathbf{x}} \in \mathcal{S}$  is such that  $\nabla_S F(\overline{\mathbf{x}}) = 0$ . Furthermore, any limit point  $\tilde{\mathbf{x}}$  of  $\{\mathbf{x}^{(k)}\}$  is such that  $\nabla_S F(\tilde{\mathbf{x}}) = 0$ .

*Proof.* Let  $\alpha_k = \delta^{j_k}$ , where  $j_k$  is given in Theorem 2. By (15) and (18) we have

$$F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k)}) \leq -\alpha_k \eta \frac{\gamma}{2} \|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|^2 \leq 0;$$

then  $\{F(\mathbf{x}^{(k)})\}$  is convergent, and the coercivity of  $F$  implies that  $\{\mathbf{x}^{(k)}\}$  is bounded. Since  $\alpha_k \geq \overline{\alpha} > 0$ , we have that

$$\lim_{k \rightarrow \infty} \|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\| = 0 \quad (19)$$

and  $\{\widehat{\mathbf{x}}^{(k)}\}$  is bounded. This, together with  $\|\overline{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\| \leq \|\overline{\mathbf{x}}^{(k)} - \widehat{\mathbf{x}}^{(k)}\| + \|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|$  and the first inequality in (13), implies that

$$\lim_{k \rightarrow \infty} \|\overline{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\| = 0 \quad (20)$$

and hence  $\{\overline{\mathbf{x}}^{(k)}\}$  is bounded. Passing to subsequences, we have

$$\lim_{k_j \rightarrow \infty} \mathbf{x}^{(k_j)} = \lim_{k_j \rightarrow \infty} \overline{\mathbf{x}}^{(k_j)} = \overline{\mathbf{x}} \in \mathcal{S}. \quad (21)$$

Since the projection onto a nonempty closed convex set is nonexpansive, we get

$$\|\nabla_{\mathcal{S}} F(\overline{\mathbf{x}}^{(k_j)})\| = \|\nabla_{\mathcal{S}} F(\overline{\mathbf{x}}^{(k_j)}) - \nabla_{\mathcal{S}} F_{k_j}(\overline{\mathbf{x}}^{(k_j)})\| \leq \|\nabla F(\overline{\mathbf{x}}^{(k_j)}) - \nabla F_{k_j}(\overline{\mathbf{x}}^{(k_j)})\|,$$

and, by using (21),

$$\lim_{k_j \rightarrow \infty} \|\nabla_{\mathcal{S}} F(\overline{\mathbf{x}}^{(k_j)})\| = \lim_{k_j \rightarrow \infty} \|\nabla F(\overline{\mathbf{x}}^{(k_j)}) - \nabla F_{k_j}(\overline{\mathbf{x}}^{(k_j)})\| = 0.$$

Then, for the lower semicontinuity of  $\|\nabla_{\mathcal{S}} F\|$ , we have

$$\nabla_{\mathcal{S}} F(\overline{\mathbf{x}}) = \mathbf{0}.$$

If  $\tilde{\mathbf{x}}$  is any limit point of  $\{\mathbf{x}^{(k)}\}$ , then  $\tilde{\mathbf{x}} \in \mathcal{S}$  and, by exploiting (20) and passing to subsequences, we have

$$\lim_{k_r \rightarrow \infty} \mathbf{x}^{(k_r)} = \lim_{k_r \rightarrow \infty} \overline{\mathbf{x}}^{(k_r)} = \tilde{\mathbf{x}} \in \mathcal{S}. \quad (22)$$

By reasoning as above we get

$$\nabla_{\mathcal{S}} F(\tilde{\mathbf{x}}) = \mathbf{0},$$

which concludes the proof.  $\square$

We note that ACQUIRE fits into the very general algorithmic framework presented in [25] and hence Theorem 3 could be derived by specializing and adapting the convergence theory of that framework, taking into account the specific properties of the functions  $D_{KL}(\mathbf{x})$  and  $TV_{\mu}(\mathbf{x})$  and their quadratic approximations  $D_{KL}^{(k)}(\mathbf{x})$  and  $TV_{\mu}^{(k)}(\mathbf{x})$ , and the line search used. However, for the sake of clarity and self-consistency, we decided to prove the convergence of Algorithm 1 from scratch.

Now we show that if the objective function is strictly convex, the whole sequence  $\{\mathbf{x}^{(k)}\}$  converges to the minimizer of problem (10).

**Theorem 4.** *Assume that the function  $F$  is strictly convex. Then the sequence  $\{\mathbf{x}^{(k)}\}$  generated by Algorithm 1 converges to a point  $\overline{\mathbf{x}} \in \mathcal{S}$  such that  $\nabla_{\mathcal{S}} F(\overline{\mathbf{x}}) = \mathbf{0}$ .*

*Proof.* We follow the line of the proof of Lemma 2 in [8]. By Theorem 3 we know that there exists a limit point  $\overline{\mathbf{x}}$  of  $\{\mathbf{x}^{(k)}\}$  such that  $\nabla_{\mathcal{S}} F(\overline{\mathbf{x}}) = \mathbf{0}$ . Since

$F$  is strictly convex,  $\bar{\mathbf{x}}$  is the optimal solution of problem (10). We must prove that  $\{\mathbf{x}^{(k)}\}$  converges to  $\bar{\mathbf{x}}$ .

From  $\alpha_k \leq 1$  it follows that  $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \|\hat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|$  and, by (19),

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| = 0.$$

Since  $\bar{\mathbf{x}}$  is a strict minimizer, there exists  $\delta > 0$  such that  $F(\bar{\mathbf{x}}) < F(\mathbf{x})$  for all  $\mathbf{x} \in \mathcal{S}$  such that  $0 < \|\mathbf{x} - \bar{\mathbf{x}}\| \leq \delta$ . For all  $\varepsilon \in (0, \delta)$ , it follows from Theorem 3 that the set  $B = \{\mathbf{x} \in \mathcal{S} : \delta \leq \|\mathbf{x} - \bar{\mathbf{x}}\| \leq \varepsilon\}$  does not contain any limit point of  $\{\mathbf{x}^{(k)}\}$ ; thus, there exists  $k_0$  such that  $\mathbf{x}^{(k)} \notin B$  for all  $k > k_0$ . Let  $k_1 \geq k_0$  such that, for all  $k > k_1$ ,

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \delta - \varepsilon.$$

Let  $K$  be the set of indices defining a subsequence of  $\{\mathbf{x}^{(k)}\}$  converging to  $\bar{\mathbf{x}}$ . There exists  $k \in K$ ,  $k > k_1$ , such that

$$\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}\| < \varepsilon,$$

and hence

$$\|\mathbf{x}^{(k+1)} - \bar{\mathbf{x}}\| \leq \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| + \|\mathbf{x}^{(k)} - \bar{\mathbf{x}}\| < \delta - \varepsilon + \varepsilon = \delta.$$

Since  $\mathbf{x}^{(k+1)} \notin B$ , we get

$$\|\mathbf{x}^{(k+1)} - \bar{\mathbf{x}}\| < \varepsilon.$$

By the same argument we can prove that  $\|\mathbf{x}^{(k+j)} - \bar{\mathbf{x}}\| < \varepsilon$  implies  $\|\mathbf{x}^{(k+j+1)} - \bar{\mathbf{x}}\| < \varepsilon$ , and hence, by induction, we have

$$\|\mathbf{x}^{(k+j)} - \bar{\mathbf{x}}\| < \varepsilon \quad \text{for all } j.$$

Since  $\varepsilon$  is arbitrary, the thesis holds.  $\square$

We conclude this section by showing that the stopping criterion (14) can be used to determine  $\hat{\mathbf{x}}^{(k)}$  at step 3 of ACQUIRE.

**Theorem 5.** *Assume that (14) holds for some  $\theta \in (0, 1)$ . Then, there exists  $\{\varepsilon_k\}$ , with  $\varepsilon_k > 0$  and  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ , such that (13) holds.*

*Proof.* First we recall that  $-\nabla F_k(\mathbf{x}) = \nabla_{\mathcal{S}} F_k(\mathbf{x}) + P_{N_{\mathcal{S}}(\mathbf{x})}(-\nabla F_k(\mathbf{x}))$  (see (9)). Since  $F_k$  is strongly convex with parameter  $\gamma$  and  $\bar{\mathbf{x}}^{(k)}$  is the solution of problem (12), we have

$$\begin{aligned} \frac{\gamma}{2} \|\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}\|^2 &\leq (\nabla F_k(\hat{\mathbf{x}}^{(k)}) - \nabla F_k(\bar{\mathbf{x}}^{(k)}))^T (\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}) \\ &= (\nabla_{\mathcal{S}} F_k(\hat{\mathbf{x}}^{(k)}))^T (\bar{\mathbf{x}}^{(k)} - \hat{\mathbf{x}}^{(k)}) + P_{N_{\mathcal{S}}(\hat{\mathbf{x}}^{(k)})}(-\nabla F_k(\hat{\mathbf{x}}^{(k)}))^T (\bar{\mathbf{x}}^{(k)} - \hat{\mathbf{x}}^{(k)}) \\ &\quad + P_{N_{\mathcal{S}}(\bar{\mathbf{x}}^{(k)})}(-\nabla F_k(\bar{\mathbf{x}}^{(k)}))^T (\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}). \end{aligned}$$

Since  $\bar{\mathbf{x}}^{(k)} - \hat{\mathbf{x}}^{(k)}$  belongs to the tangent cone at  $\hat{\mathbf{x}}^{(k)}$  and  $\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}$  belongs to the tangent cone at  $\bar{\mathbf{x}}^{(k)}$ , we get

$$\frac{\gamma}{2} \|\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}\|^2 \leq (\nabla_S F_k(\hat{\mathbf{x}}^{(k)}))^T (\bar{\mathbf{x}}^{(k)} - \hat{\mathbf{x}}^{(k)}) \leq \|\nabla_S F_k(\hat{\mathbf{x}}^{(k)})\| \|\bar{\mathbf{x}}^{(k)} - \hat{\mathbf{x}}^{(k)}\|.$$

It follows that

$$\|\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}\| \leq \frac{2}{\gamma} \|\nabla_S F_k(\hat{\mathbf{x}}^{(k)})\|;$$

thus, by requiring that

$$\|\nabla_S F_k(\hat{\mathbf{x}}^{(k)})\| \leq \theta^k \|\nabla_S F_k(\mathbf{x}^{(0)})\|$$

and setting  $\varepsilon_k = \theta^k (2/\gamma) \|\nabla_S F_k(\mathbf{x}^{(0)})\|$ , we get

$$\|\hat{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(k)}\| \leq \varepsilon_k. \quad \square$$

## 5. Numerical experiments

ACQUIRE was implemented in MATLAB, using as inner solver the scaled gradient projection (SGP) method proposed in [11], widely applied in the solution of image restoration problems. In particular, the implementation of SGP provided by the SPG-dec MATLAB code, available from <http://www.unife.it/prin/software>, was exploited.

The SGP iteration applied to problem (12) reads:

$$\mathbf{z}^{(j+1)} = \mathbf{z}^{(j)} + \rho_j \left( P_{\mathcal{S}, C_j^{-1}} \left( \mathbf{z}^{(j)} - \nu_j C_j \nabla F_k(\mathbf{z}^{(j)}) \right) - \mathbf{z}^{(j)} \right),$$

where  $\mathbf{z}^{(0)} = \mathbf{x}^{(k)}$ ,  $\rho_j$  is a line-search parameter ensuring that  $\mathbf{z}^{(j+1)}$  satisfies a sufficient decrease condition,  $\nu_j$  is a suitably chosen steplength,  $C_j$  is a diagonal positive definite matrix with diagonal entries bounded independently of  $j$ , and  $P_{\mathcal{S}, C_j^{-1}}$  is the projection operator onto  $\mathcal{S}$  with respect to the norm induced by the matrix  $C_j^{-1}$  (the dependence on  $k$  has been neglected for simplicity). Several efficient rules can be exploited to define the steplength  $\nu_j$  for the quadratic problem (12) (see, e.g., [2, 20, 21, 22, 27, 28] and the references therein). In particular, SGP uses a modification of the ABB<sub>min</sub> adaptive Barzilai-Borwein steplength defined in [28], which takes into account the scaling matrix  $C_j$  (see [11] for details); according to the analysis in [23], this steplength appears very effective. Since the steplength is computed by taking into account a certain number, say  $q$ , of suitable previous steplengths, we modified SPG-dec to avoid resetting the steplength each time the code was called, and to compute it by using  $q$  steplengths from the previous call. The diagonal scaling matrix  $C_j$  was set as in [58, section 3.3] and  $q$  was chosen equal to its default value in SPG-dec, i.e.,  $q = 3$ . The SGP iterations were stopped according to (14). For all the tests considered here, we found experimentally that  $\theta = 0.1$  worked well in the first iterations of ACQUIRE; on the other hand, criterion (14) with this

value of  $\theta$  soon becomes demanding, and fixing also a maximum number inner iterations was a natural choice. Setting this number to 10 was effective in our experiments. Defaults were used for the remaining features of **SPG-dec**.

The parameter  $\gamma$  in (11) was set equal to  $10^{-5}$ . The nonmonotone line search proposed in [31] was implemented at line 7 of Algorithm 1, with memory length equal to 5,  $\eta = 10^{-5}$ , and  $\delta = 0.5$ . **ACQUIRE** was stopped using the following criterion

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \text{Tol} \|\mathbf{x}^{(k)}\|, \quad (23)$$

i.e., when the relative change in the restored image went below a certain threshold.

**ACQUIRE** was compared with five state-of-the-art methods: **PDAL**, **SGP**, **SPIRAL-TV**, **SPLIT BREGMAN** and **VMILA**. By **PDAL** we denote our MATLAB implementation of the primal-dual algorithm proposed in [56, Algorithm 2], where we replaced the Chambolle-Pock algorithm [17] by the more efficient Primal Dual Algorithm with Linesearch introduced in [38]. Concerning the parameters of **PDAL**, following [38, Section 6] we set  $\mu = 0.7$ ,  $\delta = 0.99$  and  $\beta = 25$ . The initial steplength was chosen as  $\tau = \sqrt{2}/\omega$ , where  $\omega$  is an underestimate of  $\|M^T M\|$  and  $M = [A^T \ D_1^T \ \dots \ D_n^T]^T$  is the matrix linking the primal and dual variables. **SPIRAL-TV** is the proximal-gradient method presented in [32]; a MATLAB implementation of it is available from <http://drz.ac/code/>. By **SPLIT BREGMAN** we denote a version of the method proposed in [30], which was specialized for problem (4) [29] and implemented in the MATLAB code **tvdeconv** available from <http://dev.ipol.im/~getreuer/code/>. Finally, **VMILA** is the variable-metric inexact line-search proximal-gradient method described in [9], whose MATLAB implementation can be found at <http://www.oasis.unimore.it/site/home/software>. In all the methods, the stopping criterion (23) was applied. **SGP** was run with the same setting of parameters used to solve the subproblems in **ACQUIRE**. For **SPIRAL-TV**, **SPLIT BREGMAN** and **VMILA**, the default values of the parameters were used.

**PDAL**, **SPIRAL-TV**, **SPLIT BREGMAN** and **VMILA** do not require any smooth approximation of TV and were run directly on problem (4). Therefore, our comparison also provides some insight into the effects of using a smoothed version of TV. **ACQUIRE** was run with and without the flux constraint, i.e., using both feasible sets  $\mathcal{S}_1$  and  $\mathcal{S}_2$  – see (5) and (6). However, since the use of the flux constraint did not lead to any significant difference in the restored images, and this constraint was not available in the implementations of **SPIRAL-TV**, **SPLIT BREGMAN** and **VMILA**, we report only the results for  $\mathcal{S} = \mathcal{S}_1$ .

As already noted, when the matrix  $A$  represents a convolution, the matrix-vector products involving the matrices  $A$  and  $A^T$  can be performed by using fast algorithms. This is the case for all the experiments considered in this work. Since periodic boundary conditions were considered for all the images used as test problems, the matrix-vector products were performed by exploiting the MATLAB FFT functions **fft2** and **ifft2**.

In order to build the test problems used in the experiments, four reference images were chosen: *cameraman*, *micro*, *phantom* and *satellite*, shown in Fig-



ure 1. The cameraman image, available in the MATLAB Image Processing Toolbox, is widely used in the literature since it contains both sharp edges and flat regions and presents a nice mixture of smooth and nonsmooth regions; micro is the confocal microscopy phantom described in [57]; phantom is the famous Shepp-Logan brain phantom described in [50]; finally, the satellite image comes from the RestoreTools package [41]. The size of cameraman, phantom and satellite is  $256 \times 256$ , while the size of micro is  $128 \times 128$ .

A first set of test problems, T1, was obtained by convolving each reference image with a Gaussian PSF and corrupting the resulting image with Poisson noise. A further set of test problems, T2, was built by convolving some of the images with a motion blur PSF and an out-of-focus PSF, and then introducing Poisson noise. Details about the PSFs and the Poisson noise are given in Subsections 5.1 and 5.2, where the results of numerical experiments performed by using the corrupted images are also reported.

All the experiments were carried out on a 2.5 GHz Intel Core i7 processors with 16 GB of RAM, 4 MB of L3 cache and the macOS 10.13.6 operating system, using MATLAB R2018b.

### 5.1. Results on images with Gaussian blur

The Gaussian blur PSF for constructing the test set T1 was computed by using the function `psfGauss` from [41], choosing the variance  $\sigma$  as specified in Table 1. In order to take into account the existence of some background emission,  $10^{-10}$  was added to all the pixels of the blurred image; obviously, the vector  $\mathbf{b}$  in  $D_{KL}(\mathbf{x})$  was set as  $\mathbf{b} = 10^{-10}\mathbf{e}$ . The Poisson noise was introduced with the function `imnoise` from the MATLAB Image Processing Toolbox. Note that for this type of noise, which affects the photon counting process, the Signal-to-Noise Ratio (SNR) is usually estimated by

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

where  $N_{\text{exact}}$  and  $N_{\text{background}}$  are the total number of photons in the exact image to be recovered and in the background term, respectively. Therefore, in order to obtain noisy and blurred images with SNR equal to 35 and 40, the intensities of the reference images were suitably pre-scaled. The resulting images are shown in Figures 2-5 (left columns).

The regularization parameter  $\lambda$  was set by trial and error, as described next. The search for a suitable value of  $\lambda$  was carried out separately for the minimization problem (4), which uses the original TV, and the minimization problem (10), which uses the smoothed TV. In the latter case, ACQUIRE was run several times on each test image, for 25 seconds and with  $\text{Tol} = 0$  (see (23)), slowly varying the value of  $\lambda$  at each execution. The value of  $\lambda$  corresponding to the smallest relative error at the last iteration, was chosen to perform the experiments with ACQUIRE and SGP. Note that, by running SGP for more than 25 seconds, we also verified that the selected value of  $\lambda$  was suited to SGP

too. The relative error was computed as  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|/\|\mathbf{x}^*\|$ , where  $\mathbf{x}^*$  denotes the original image. The values of  $\lambda$  for problem (4) were set using the same strategy applied to (10). In this case, instead of ACQUIRE, for each image we used the method that appeared more efficient among PDAL, SPIRAL-TV, SPLIT-BREGMAN and VMILA, on the basis of preliminary experiments. All the values of  $\lambda$  are reported in Table 1. The same value of the regularization parameter was determined for both the original and the smoothed TV, except for the satellite image; however, very close values of  $\lambda$  were obtained in this case.

The parameter  $\mu$  in the smoothed version of TV was set as  $\mu = 10^{-2}$ , which, by numerical experiments, appeared to achieve a good tradeoff between approximation accuracy and computational effort, for all the test problems. Indeed, as the value of  $\mu$  decreases,  $TV_\mu$  becomes a more accurate approximation of  $TV$ , but the condition number of its Hessian increases. For both versions of TV, each corrupted image was scaled by division by its largest intensity value before applying any method; the scaled image was also used as starting guess, i.e., we set  $\mathbf{x}^{(0)} = \mathbf{y}$ . We also performed experiments by setting  $\mathbf{x}^{(0)}$  as the vector with entries equal to the flux of the scaled image divided by the number of pixels of the image, but we could not see any significant difference in the results.

ACQUIRE was compared with all the other methods on the test problems previously described, in terms of accuracy and execution time. Six values of Tol were considered,  $\text{Tol} = 10^{-2}, 10^{-3}, \dots, 10^{-7}$ , with the aim of assessing the behavior of ACQUIRE with different accuracy requirements and getting useful information for the effective use of an automatic stopping rule. A maximum execution time of 25 seconds was also set for all the methods.

Figures 6 and 7 show the relative errors and the execution times of each method, in seconds, versus the stopping tolerances, for the problems with  $\text{SNR} = 35$  and  $\text{SNR} = 40$ , respectively. The images obtained with ACQUIRE and corresponding to the smallest errors are shown in Figures 2-5 (right columns). Further details concerning all the methods are given in Tables 3 and 4, where we report the smallest errors, the iterations performed to achieve them, the corresponding execution times and tolerances, and the values of MSSIM for the restored images corresponding to the smallest errors. MSSIM is a structural similarity measure index [54] which is related to the perceived visual quality of the image; the higher its value, the better the perceived similarity between the restored and original images.

We see that ACQUIRE generally does not need small tolerances to achieve small errors, because of its fast progress in the first iterations, which produces large changes in the iterate. We note that in four test cases it reaches its minimum error with  $\text{Tol} = 10^{-3}$ ; this is consistent with the exploitation of second-order information to build the quadratic model at each iteration. SGP generally achieves errors comparable with those of ACQUIRE, but its progress at each iteration is slower, and hence it often requires smaller tolerances to avoid stopping prematurely. On the other hand, a single iteration of ACQUIRE requires more time than an iteration of SGP, and the former method may be

either faster or slower than the latter in obtaining small errors. PDAL is able to achieve errors comparable with those of ACQUIRE, but it generally requires smaller tolerances and larger times. VMILA is very efficient on both instances of the cameraman problem and on the phantom problem with  $\text{SNR} = 35$ , where it is faster than ACQUIRE or comparable with it. However, there are some problems where VMILA makes very little progress in the first iterations, leading to very premature stops, as shown by the almost constant execution times in the pictures. The remaining methods are generally less efficient than the previous ones, because of their very slow progress in reducing the error. We note that the errors of ACQUIRE and SGP show a light semiconvergence for some problems. We were not able to completely remove this behavior by increasing the regularization parameter without significantly deteriorating the visual quality of the image and decided to keep the value of  $\lambda$  determined by the procedure previously described. Finally, we note that the values of MSSIM corresponding to ACQUIRE confirm that in most cases this method is able to provide better or similar quality images in comparison with the other methods.

## 5.2. Results on images with moving and out-of-focus blurs

In order to understand if the previous behavior of ACQUIRE also holds for blurs different from the Gaussian one, and to further compare ACQUIRE with the other methods, we built the test set T2. We introduced in two reference images, cameraman and satellite, the motion blur and the out-of-focus blur, which simulate the linear motion of a camera and the out-of-focus effect, respectively. Both blurs were computed by using the Matlab function `fspecial`. Poisson noise with SNR equal to 35 and 40 was introduced in the blurred images, as in the case of Gaussian blur. The length and the angle (in degrees) of the motion, `len` and  $\varphi$ , and the radius of the disk kernel for the out-of-focus effect, `rad`, are specified in Table 2. The values of  $\lambda$ , obtained with the procedure described in Subsection 5.1, are reported in the same table. In this case, small differences can be observed between the values of the regularization parameter corresponding to the original and the smoothed TV. The parameter  $\mu$  in the smoothed version of TV was set again as  $\mu = 10^{-2}$ , on the basis of numerical experiments. Each noisy and blurred image was scaled as in the case of Gaussian blur. The vector with entries equal to the flux of the scaled image divided by the number of pixels of the image was used as starting guess, because, with the motion and out-of-focus blurs, this choice generally appeared more effective than the choice of the scaled image.

The error and time histories shown in Figures 12 and 13 confirm the behavior of ACQUIRE: it is able to strongly reduce the error in the first iterations and hence, although its single iteration is usually more expensive than a single iterations of the other methods, it is competitive with those methods. Furthermore, ACQUIRE allows an effective use of an automatic stopping rule, avoiding premature stopping, which may happen with other methods. This is confirmed by the data in Tables 5 and 6, which report, for all the methods, the smallest errors and the corresponding MSSIM values, number of iterations, execution

times and tolerances. The images restored by ACQUIRE and corresponding to the smallest errors are shown in the right columns of Figures 8-11).

## 6. Conclusions

We proposed ACQUIRE, a method for TV-based restoration of images corrupted by Poisson noise, modeled by (4). ACQUIRE is a line-search method which considers a smoothed version of TV and computes the search directions by minimizing quadratic models built by exploiting second-order information about the objective function, which is usually not taken into account in methods for problem (4). We proved that the sequence generated by our method has a subsequence converging to a minimizer of the smoothed problem (10) and that any limit point is a minimizer; furthermore, if the problem is strictly convex, the whole sequence is convergent. We note that convergence holds without requiring the exact minimization of the quadratic models; low accuracy in this minimization can be used in practice, as shown by the numerical results.

Computational experiments on reference test cases showed that the exploitation of second-order information is beneficial, since it generally leads to a significant reduction of the reconstruction error in the first iterations. Furthermore, the capability of achieving a tradeoff between accuracy and efficiency makes ACQUIRE competitive with well-established methods for TV-based Poisson image restoration.

Finally, we observe that our approach can be extended to other regularization models, such as high-order Total Variation [36, 43] and Total Generalized Variation [12, 13], which have been proposed to reduce the staircase effect of TV and retain the fine details of the image.

## Acknowledgments

We wish to thank the anonymous reviewers for their insightful comments and useful suggestions, which helped us improve the quality of our work.

## References

- [1] R.-C. BARNARD, H. BILHEUX, AND T. E. A. TOOPS, *Total variation-based neutron computed tomography*, Review of Scientific Instruments, 89 (2018), 053704.
- [2] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 8 (1988), pp. 141–148.
- [3] A. BECK AND M. TEOULLE, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Transactions on Image Processing, 18 (2009), pp. 2419–2434.

- [4] M. BERTERO, P. BOCCACCI, G. DESIDERÀ, AND G. VICIDOMINI, *Image deblurring with Poisson data: from cells to galaxies*, Inverse Problems, 25 (2009), 123006.
- [5] M. BERTERO, P. BOCCACCI, G. TALENTI, R. ZANELLA, AND L. ZANNI, *A discrepancy principle for poisson data*, Inverse Problems, 26 (2010), 105004.
- [6] M. BERTERO, H. LANTÉRI, AND L. ZANNI, *Iterative image reconstruction: a point of view*, in “Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT)”, pp. 37–63 (Y. Censor, M. Jiang and A. K. Louis eds.), CRM Series, Edizioni della Normale, 2008.
- [7] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, Belmont, MA, USA, 1999.
- [8] E. G. BIRGIN, N. KREJIĆ, AND J. M. MARTÍNEZ, *Globally convergent inexact quasi-Newton methods for solving nonlinear systems*, Numerical Algorithms, 32 (2003), pp. 249–260.
- [9] S. BONETTINI, I. LORIS, F. PORTA, AND M. PRATO, *Variable metric inexact line-search-based methods for nonsmooth optimization*, SIAM Journal on Optimization, 26 (2016), pp. 891–921.
- [10] S. BONETTINI AND V. RUGGIERO, *An alternating extragradient method for total variation-based image restoration from Poisson data*, Inverse Problems, 27 (2011), 095001.
- [11] S. BONETTINI, R. ZANELLA, AND L. ZANNI, *A scaled gradient projection method for constrained image deblurring*, Inverse Problems, 25 (2009), 015002.
- [12] K. BREDIES AND M. HOLLER, *Regularization of linear inverse problems with total generalized variation*, Journal of Inverse and Ill-posed Problems, 22 (2014), pp. 871–913.
- [13] K. BREDIES, K. KUNISCH, AND T. POCK, *Total generalized variation*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 492–526.
- [14] P. H. CALAMAI AND J. J. MORÉ, *Projected gradient methods for linearly constrained problems*, Mathematical Programming, 39 (1987), pp. 93–116.
- [15] P. H. CALAMAI AND J. J. MORÉ, *Quasi-Newton updates with bounds*, SIAM Journal on Numerical Analysis, 24 (1987), pp. 1434–1441.
- [16] A. CHAMBOLLE, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97.
- [17] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.

- [18] L. CONDAT, *Fast projection onto the simplex and the  $l_1$  ball*, Mathematical Programming, Series A, 158 (2016), pp. 575–585.
- [19] Y.-H. DAI AND R. FLETCHER, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Mathematical Programming, Series A, 106 (2006), pp. 403–421.
- [20] R. DE ASMUNDIS, D. DI SERAFINO, W. W. HAGER, G. TORALDO, AND H. ZHANG, *An efficient gradient method using the Yuan steplength*, Computational Optimization and Applications, 59 (2014), pp. 541–563.
- [21] R. DE ASMUNDIS, D. DI SERAFINO, AND G. LANDI, *On the regularizing behavior of the SDA and SDC gradient methods in the solution of linear ill-posed problems*, Journal of Computational and Applied Mathematics, 302 (2016), pp. 81–93.
- [22] R. DE ASMUNDIS, D. DI SERAFINO, F. RICCIO, AND G. TORALDO, *On spectral properties of steepest descent methods*, IMA Journal of Numerical Analysis, 33 (2013), pp. 1416–1435.
- [23] D. DI SERAFINO, V. RUGGIERO, G. TORALDO, AND L. ZANNI, *On the steplength selection in gradient methods for unconstrained optimization*, Applied Mathematics and Computation, 318 (2018), pp. 176–195.
- [24] D. DI SERAFINO, G. TORALDO, M. VIOLA, AND J. BARLOW, *A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables*, SIAM Journal on Optimization, 28 (2018), pp. 2809–2838.
- [25] F. FACCHINEI, L. LAMPARIELLO, AND G. SCUTARI, *Feasible methods for nonconvex nonsmooth problems with applications in green communications*, Mathematical Programming, 164 (2017), pp. 55–90.
- [26] M. A. T. FIGUEIREDO AND J. M. BIOUSCAS-DIAS, *Restoration of Poissonian images using alternating direction optimization*, IEEE Transactions on Image Processing, 19 (2010), pp. 3133–3145.
- [27] R. FLETCHER, *A limited memory steepest descent method*, Mathematical Programming, Series A, 135 (2012), pp. 413–436.
- [28] G. FRASSOLDATI, L. ZANNI, AND G. ZANGHIRATI, *New adaptive step-size selections in gradient methods*, Journal of Industrial and Management Optimization, 4 (2008), pp. 299–312.
- [29] P. GETREUER, *Total variation deconvolution using split Bregman*, Image Processing On Line, 2 (2012), pp. 158–174.
- [30] T. GOLDSTEIN AND S. OSHER, *The split Bregman method for  $L1$ -regularized problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 323–343.

- [31] L. GRIPPO, F. LAMPARIELLO, AND S. LUCIDI, *A nonmonotone line search technique for Newton's method*, SIAM Journal on Numerical Analysis, 23 (1986), pp. 707–716.
- [32] Z. T. HARMANY, R. F. MARCIA, AND R. M. WILLETT, *This is SPIRAL-TAP: Sparse Poisson Intensity Reconstruction Algorithms – Theory and Practice*, IEEE Transactions on Image Processing, 21 (2012), pp. 1084–1096.
- [33] G. HERMAN, *Fundamentals of Computerized Tomography*, Springer, Berlin, 2009.
- [34] E. JONSSON, S.-C. HUANG, AND T. CHAN, *Total Variation regularization in positron emission tomography*, Tech. Report 98–48, UCLA, CAM-Report, 1998.
- [35] Z. LI, F. MALGOUYRES, AND T. ZENG, *Regularized non-local Total Variation and application in image restoration* Journal of Mathematical Imaging and Vision, 59 (2017), pp. 296–317.
- [36] J. LIU, T.-Z. HUANG, X.-G. LV, AND S. WANG, *High-order total variation-based Poissonian image deconvolution with spatially adapted regularization parameter*, Applied Mathematical Modelling, 45 (2017), pp. 516–529.
- [37] E. LOLI PICCOLOMINI, V. L. COLI, E. MOROTTI, AND L. ZANNI, *Reconstruction of 3D X-ray CT images from reduced sampling by a scaled gradient projection algorithm*, Computational Optimization and Applications, 71 (2018), pp. 171–191.
- [38] Y. MALITSKY AND T. POCK, *A first-order primal-dual algorithm with linesearch*, SIAM Journal on Optimization, 28 (2018), pp. 411–432.
- [39] J. J. MORÉ AND G. TORALDO, *On the solution of large quadratic programming problems with bound constraints*, SIAM Journal on Optimization, 1 (1991), pp. 93–113.
- [40] A.-M. MOTA, N. MATELA, N. OLIVERA, AND P. ALMEIDA, *3D total variation minimization filter for breast tomosynthesis imaging*, Proc. of 13th International Workshop IWDM, (2016), pp. 484–492.
- [41] J. G. NAGY, K. PALMER, AND L. PERRONE, *Iterative methods for image deblurring: a Matlab object-oriented approach*, Numerical Algorithms, 36 (2004), pp. 73–93.
- [42] V. Y. PANIN, G. L. ZENG, AND G. T. GULLBERG, *Total variation regulated EM algorithm*, in 1998 IEEE Nuclear Science Symposium Conference Record, 1998, p. 6359010.

- [43] K. PAPAITSOROS AND C.-B. SCHÖNLIEB, *A combined first and second order variational approach for image reconstruction*, Journal of Mathematical Imaging and Vision, 48 (2014), pp. 308–338.
- [44] J. B. PAWLEY, *Handbook of Biological Confocal Microscopy*, Plenum Press, New York, 2nd ed., 1996.
- [45] P. RODRÍGUEZ AND B. WOHLBERG, *Efficient minimization method for a generalized total variation functional*, IEEE Transactions on Image Processing, 18 (2009), pp. 322–332.
- [46] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.
- [47] P. SARDER AND A. NEHORAI, *Deconvolution method for 3-D fluorescence microscopy images*, IEEE Signal Processing Letters, 23 (2006), pp. 32–45.
- [48] A. SAWATZKY, C. BRUNE, F. WÜBBELING, T. KÖSTERS, K. SCHÄFERS, AND M. BURGER, *Accurate EM-TV algorithm in PET with low SNR*, in 2008 IEEE Nuclear Science Symposium Conference Record, 2008, <https://doi.org/10.1109/NSSMIC.2008.4774392>.
- [49] S. SETZER, G. STEIDL, AND T. TEUBER, *Deblurring Poissonian images by split Bregman techniques*, Journal of Visual Communication and Image Representation, 21 (2010), pp. 193–199.
- [50] L. A. SHEPP AND B. LOGAN, *The Fourier reconstruction of a head section*, IEEE Transactions on Nuclear Science, 21 (1974), pp. 21–43.
- [51] L. A. SHEPP AND Y. VARDI, *Maximum likelihood reconstruction for emission tomography*, IEEE Transactions on Medical Imaging, 1 (1982), pp. 113–122.
- [52] J.-L. STARCK AND F. MURTAGH, *Astronomical Image and Data Analysis*, Springer, 2006.
- [53] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM Journal on Scientific Computing, 17 (1996), pp. 227–238.
- [54] Z. WANG, A. C. BOVIK, H. R. SHEIKH, AND E. P. SIMONCELLI, *Image quality assessment: from error visibility to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612.
- [55] P. WEISS, L. BLANC-FÉRAUD, AND G. AUBERT, *Efficient schemes for total variation minimization under constraints in image processing*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2047–2080.
- [56] Y.-W. WEN, R. H. CHAN, AND T.-Y. ZENG, *Primal-dual algorithms for total variation based image restoration under poisson noise*, Science China Mathematics, 59 (2016), pp. 141–160.



- [57] R. M. WILLETT AND R. D. NOWAK, *Platelets: a multiscale approach for recovering edges and surfaces in photon limited medical imaging*, IEEE Transactions on Medical Imaging, 22 (2003), pp. 332–350.
- [58] R. ZANELLA, P. BOCCACCI, L. ZANNI, AND M. BERTERO, *Efficient gradient projection methods for edge-preserving removal of Poisson noise*, Inverse Problems, 25 (2009), 045010.
- [59] E. H. ZARANTONELLO, *Projections on convex sets in Hilbert space and spectral theory*, in Contributions to Nonlinear Functional Analysis, E. H. Zarantonello, ed., Academic Press, New York, NY, USA, 1971.
- [60] J. ZHANG, Y. HU, AND J. NAGY, *A scaled gradient method for digital tomographic image reconstruction*, Inverse Problems and Imaging, 12 (2018), pp. 239–259.

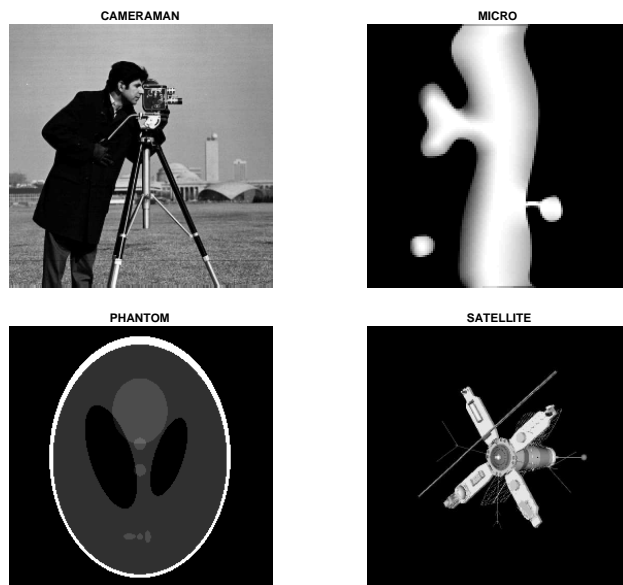


Figure 1: Reference images: cameraman, micro, phantom and satellite.



Figure 2: Cameraman: images corrupted by Gaussian blur and Poisson noise (left) and images restored by ACQUIRE (right).

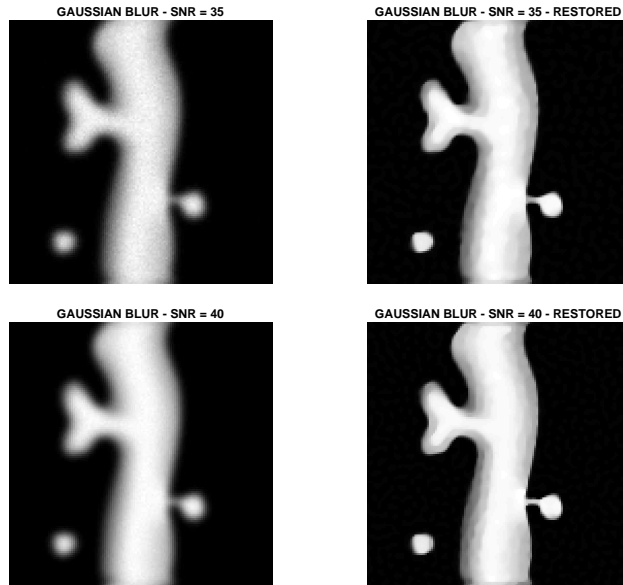


Figure 3: Micro: images corrupted by Gaussian blur and Poisson noise (left) and images restored by ACQUIRE (right).

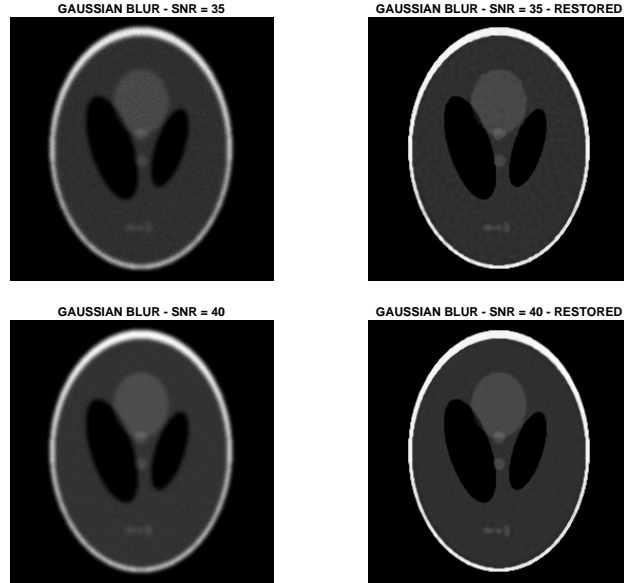


Figure 4: Phantom: images corrupted by Gaussian blur and Poisson noise (left) and images restored by ACQUIRE (right).

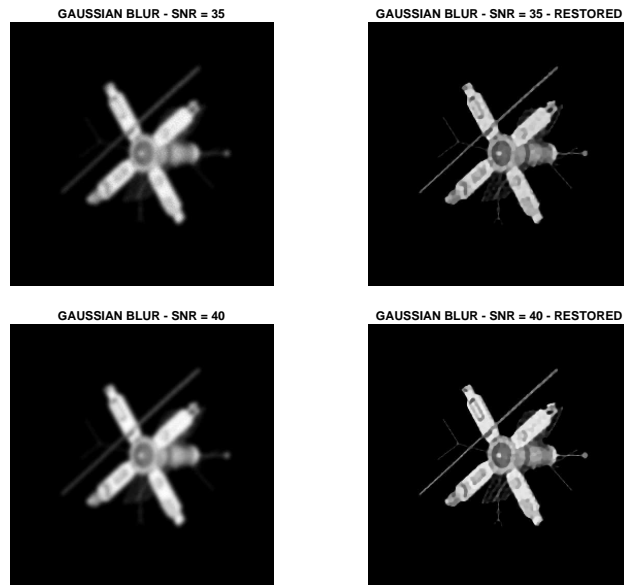


Figure 5: Satellite: images corrupted by Gaussian blur and Poisson noise (left) and images restored by ACQUIRE (right).

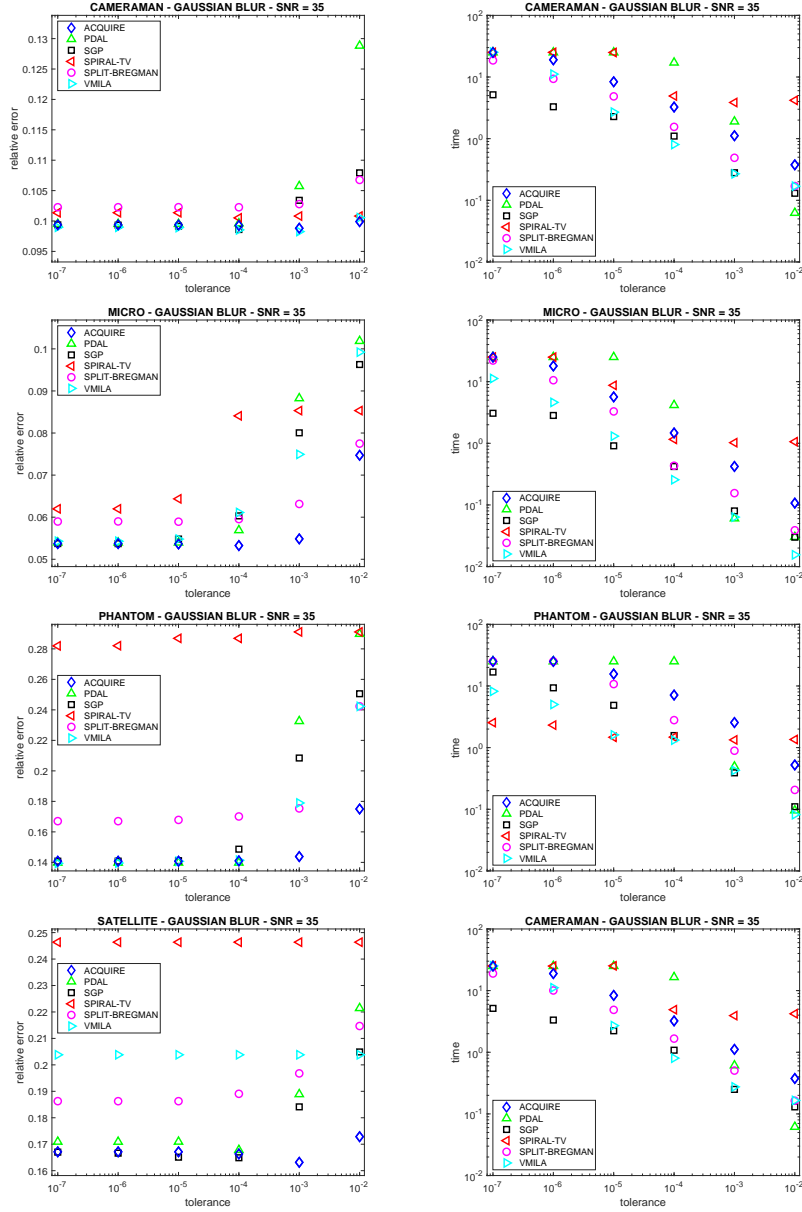


Figure 6: Test set T1, SNR = 35: relative error (left) and execution time (right) versus tolerance, for all the methods.

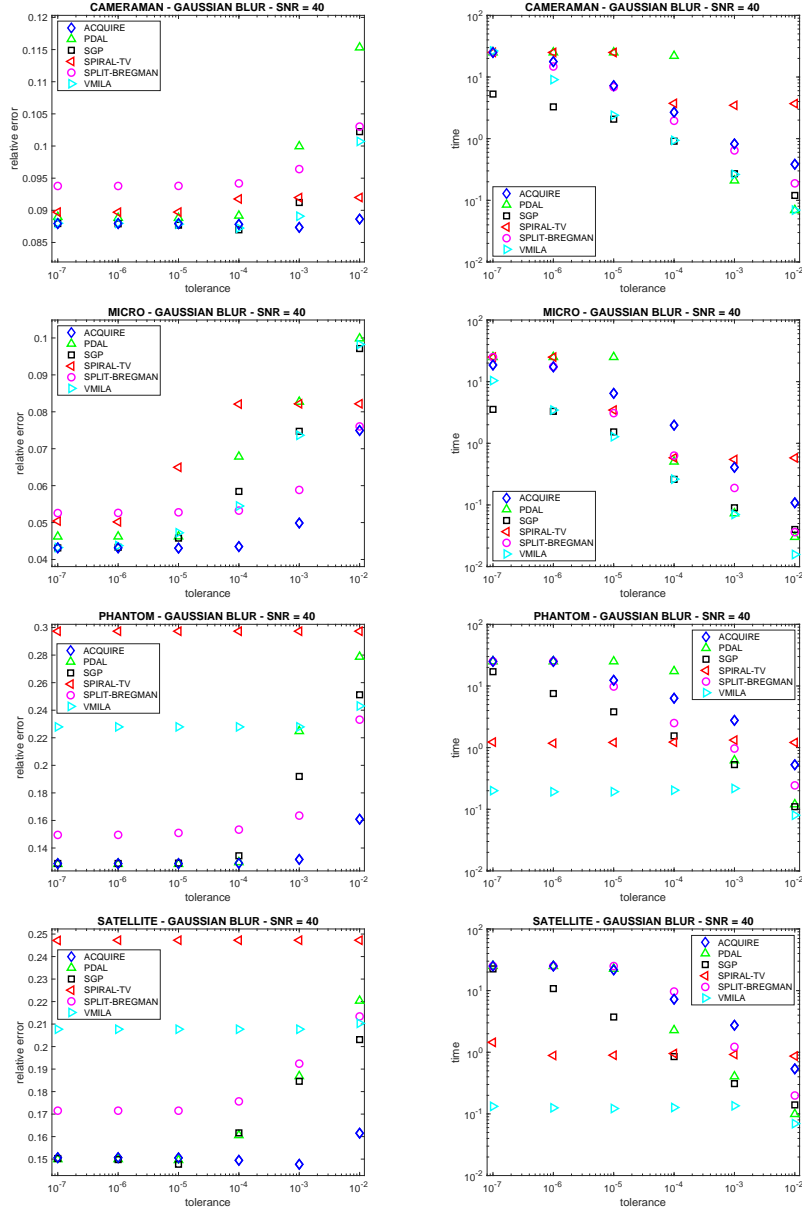


Figure 7: Test set T1, SNR = 40: relative error (left) and execution time (right) versus tolerance, for all the methods.

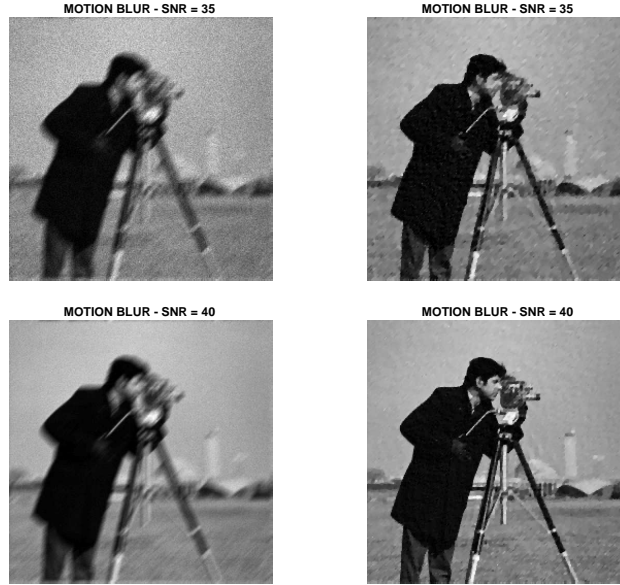


Figure 8: Cameraman: images corrupted by motion blur and Poisson noise (left) and images restored by ACQUIRE (right).

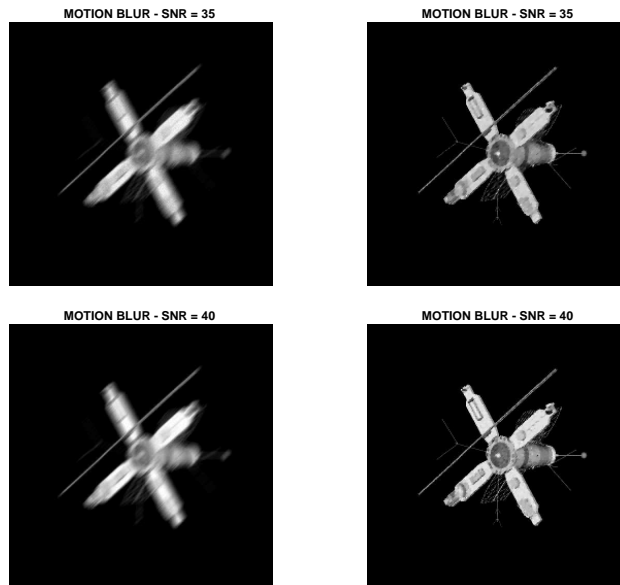


Figure 9: Satellite: images corrupted by motion blur and Poisson noise (left) and images restored by ACQUIRE (right).

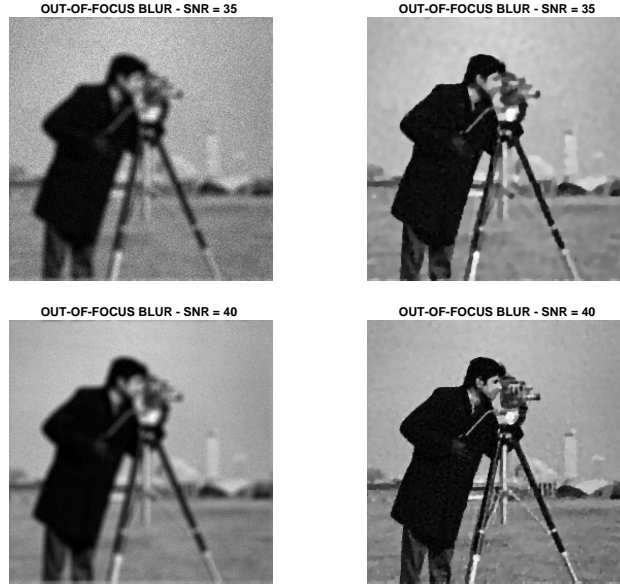


Figure 10: Cameraman: images corrupted by out-of-focus blur and Poisson noise (left) and images restored by ACQUIRE (right).

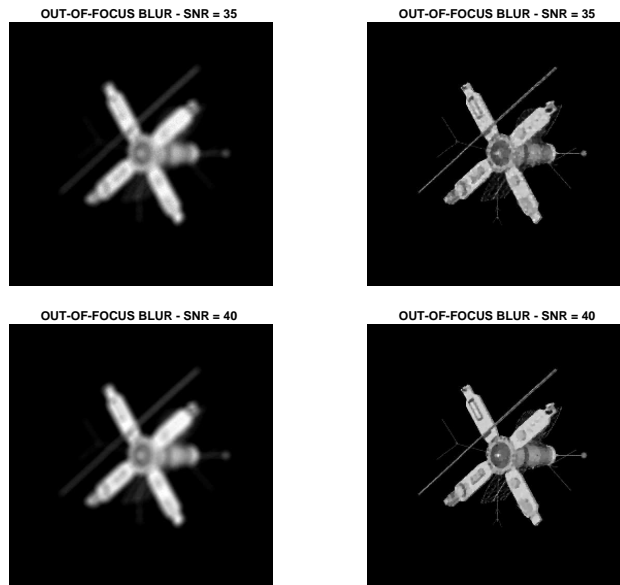


Figure 11: Satellite: images corrupted by out-of-focus blur and Poisson noise (left) and images restored by ACQUIRE (right).



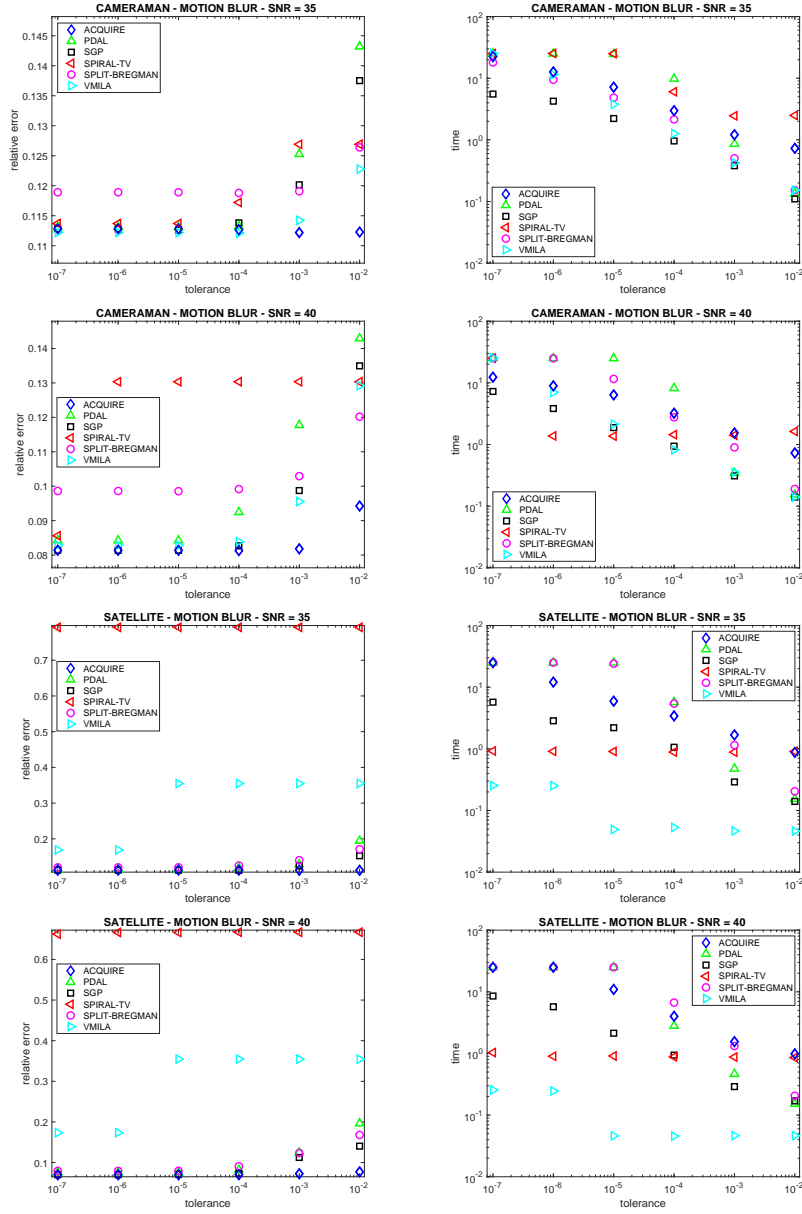


Figure 12: Test set T2, motion blur, SNR = 35, 40: relative error (left) and execution time (right) versus tolerance, for all the methods.

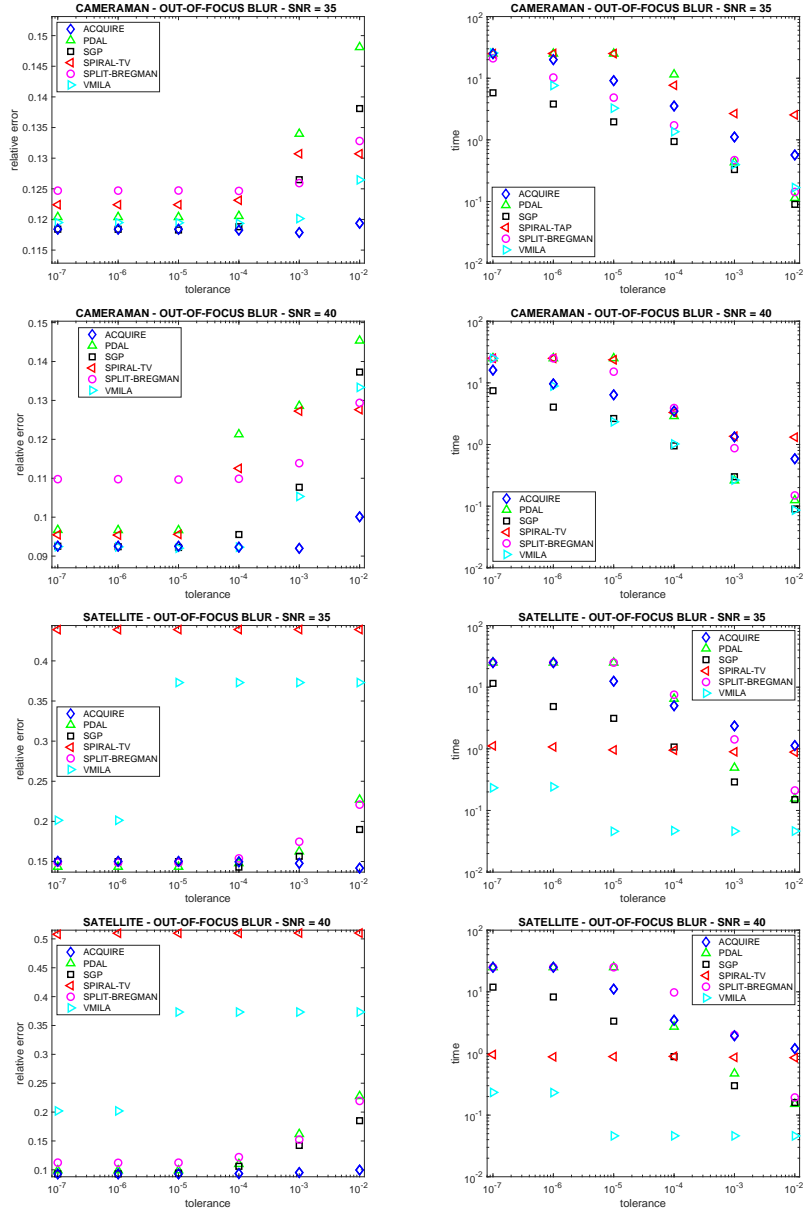


Figure 13: Test set T2, out-of-focus blur, SNR = 35,40: relative error (left) and execution time (right) versus tolerance, for all the methods.

Problem	$\sigma$	SNR	$\lambda$	
			$TV$	$TV_\mu$
cameraman	1.4	35	1.55e-2	1.55e-2
		40	5.00e-3	5.00e-3
micro	2.0	35	4.50e-3	4.50e-3
		40	1.00e-3	1.00e-3
phantom	2.0	35	6.00e-3	6.00e-3
		40	4.00e-3	4.00e-3
satellite	2.0	35	7.00e-4	9.00e-4
		40	1.50e-4	9.00e-5

Table 1: Details of test set T1.

motion blur				
Problem	(len, $\varphi$ )	SNR	$\lambda$	
			$TV$	$TV_{\mu}$
cameraman	(11,45)	35	1.50e-2	0.75e-2
		40	2.50e-3	1.75e-3
satellite	(11,45)	35	2.00e-3	1.50e-3
		40	3.50e-4	2.25e-4
out-of-focus blur				
Problem	rad	SNR	$\lambda$	
			$TV$	$TV_{\mu}$
cameraman	4	35	1.50e-2	1.00e-2
		40	1.40e-3	1.20e-3
satellite	4	35	1.75e-3	0.50e-3
		40	2.75e-4	1.90e-4

Table 2: Details of test set T2.

Test set T1, SNR = 35					
Method	Min rel err	MSSIM	Iters	Time	Tol
cameraman					
ACQUIRE	9.88e−2	8.01e−1	11	1.12e+0	1.00e−3
PDAL	9.95e−2	8.03e−1	2743	2.50e+1	1.00e−5
SGP	9.86e−2	8.01e−1	67	1.10e+0	1.00e−4
SPIRAL-TV	1.01e−1	7.99e−1	62	4.92e+0	1.00e−4
SPLIT-BREGMAN	1.02e−1	8.04e−1	116	1.55e+0	1.00e−4
VMILA	9.83e−2	8.00e−1	18	2.69e−1	1.00e−3
micro					
ACQUIRE	5.33e−2	9.72e−1	50	1.47e+0	1.00e−4
PDAL	5.40e−2	9.72e−1	9974	2.50e+1	1.00e−5
SGP	5.37e−2	9.72e−1	361	2.83e+0	1.00e−6
SPIRAL-TV	6.20e−2	9.72e−1	1081	2.50e+1	1.00e−7
SPLIT-BREGMAN	5.89e−2	9.77e−1	1107	3.29e+0	1.00e−5
VMILA	5.43e−2	9.76e−1	501	4.61e+0	1.00e−6
phantom					
ACQUIRE	1.41e−1	9.75e−1	220	2.50e+1	1.00e−6
PDAL	1.40e−1	9.73e−1	2754	2.50e+1	1.00e−6
SGP	1.41e−1	9.75e−1	769	1.68e+1	1.00e−7
SPIRAL-TV	2.82e−1	9.22e−1	106	2.54e+0	1.00e−7
SPLIT-BREGMAN	1.67e−1	9.73e−1	2006	2.50e+1	1.00e−6
VMILA	1.39e−1	9.80e−1	483	8.20e+0	1.00e−7
satellite					
ACQUIRE	1.63e−1	9.62e−1	20	1.89e+0	1.00e−3
PDAL	1.68e−1	9.61e−1	275	2.52e+0	1.00e−4
SGP	1.65e−1	9.61e−1	84	1.36e+0	1.00e−4
SPIRAL-TV	2.46e−1	9.11e−1	51	8.49e−1	1.00e−2
SPLIT-BREGMAN	1.86e−1	9.45e−1	1985	2.50e+1	1.00e−5
VMILA	2.04e−1	9.40e−1	9	7.73e−2	1.00e−2

Table 3: Test set T1, SNR = 35: minimum relative error achieved by each method and corresponding MSSIM value, number of iterations, execution time and tolerance.

Test set T1, SNR = 40					
Method	Min rel err	MSSIM	Iters	Time	Tol
cameraman					
ACQUIRE	8.73e−2	8.42e−1	8	8.23e−1	1.00e−3
PDAL	8.88e−2	8.22e−1	2733	2.50e+1	1.00e−5
SGP	8.70e−2	8.42e−1	53	9.10e−1	1.00e−4
SPIRAL-TV	8.97e−2	8.36e−1	280	2.50e+1	1.00e−6
SPLIT-BREGMAN	9.38e−2	8.41e−1	1962	2.50e+1	1.00e−7
VMILA	8.72e−2	8.42e−1	58	9.39e−1	1.00e−4
micro					
ACQUIRE	4.31e−2	9.82e−1	218	6.46e+0	1.00e−5
PDAL	4.62e−2	9.64e−1	9543	2.50e+1	1.00e−7
SGP	4.31e−2	9.81e−1	700	3.55e+0	1.00e−7
SPIRAL-TV	5.02e−2	9.83e−1	1480	2.50e+1	1.00e−6
SPLIT-BREGMAN	5.26e−2	9.85e−1	7766	2.50e+1	1.00e−7
VMILA	4.32e−2	9.85e−1	1223	1.04e+1	1.00e−7
phantom					
ACQUIRE	1.29e−1	9.85e−1	217	2.50e+1	1.00e−6
PDAL	1.28e−1	9.79e−1	2650	2.50e+1	1.00e−5
SGP	1.29e−1	9.85e−1	369	7.54e+0	1.00e−6
SPIRAL-TV	2.97e−1	9.10e−1	51	1.21e+0	1.00e−2
SPLIT-BREGMAN	1.50e−1	9.83e−1	1936	2.50e+1	1.00e−7
VMILA	2.28e−1	9.51e−1	16	2.18e−1	1.00e−3
satellite					
ACQUIRE	1.48e−1	9.70e−1	28	2.74e+0	1.00e−3
PDAL	1.50e−1	9.69e−1	2417	2.26e+1	1.00e−5
SGP	1.48e−1	9.70e−1	216	3.73e+0	1.00e−5
SPIRAL-TV	2.47e−1	9.11e−1	85	1.45e+0	1.00e−7
SPLIT-BREGMAN	1.72e−1	9.53e−1	1935	2.50e+1	1.00e−5
VMILA	2.08e−1	9.37e−1	10	1.35e−1	1.00e−3

Table 4: Test set T1, SNR = 40: minimum relative error achieved by each method and corresponding MSSIM value, number of iterations, execution time and tolerance.

Test set T2, motion blur					
Method	Min rel err	MSSIM	Iters	Time	Tol
cameraman, SNR = 35					
ACQUIRE	1.12e−1	6.85e−1	12	1.21e+0	1.00e−3
PDAL	1.13e−1	7.51e−1	2656	2.50e+1	1.00e−6
SGP	1.13e−1	6.77e−1	110	2.22e+0	1.00e−5
SPIRAL-TV	1.14e−1	7.54e−1	242	2.51e+1	1.00e−5
SPLIT-BREGMAN	1.19e−1	7.50e−1	164	2.15e+0	1.00e−4
VMILA	1.12e−1	7.51e−1	73	1.27e+0	1.00e−4
cameraman, SNR = 40					
ACQUIRE	8.13e−2	8.09e−1	31	3.21e+0	1.00e−4
PDAL	8.42e−2	8.04e−1	2698	2.50e+1	1.00e−5
SGP	8.14e−2	8.09e−1	219	3.83e+0	1.00e−6
SPIRAL-TV	8.56e−2	8.25e−1	377	2.52e+1	1.00e−7
SPLIT-BREGMAN	9.85e−2	8.06e−1	880	1.16e+1	1.00e−5
VMILA	8.28e−2	8.27e−1	139	2.14e+0	1.00e−5
satellite, SNR = 35					
ACQUIRE	1.12e−1	9.81e−1	9	8.76e−1	1.00e−2
PDAL	1.14e−1	9.81e−1	2680	2.50e+1	1.00e−5
SGP	1.12e−1	9.81e−1	64	1.06e+0	1.00e−4
SPIRAL-TV	7.92e−1	8.78e−1	51	9.09e−1	1.00e−2
SPLIT-BREGMAN	1.20e−1	9.77e−1	2099	2.50e+1	1.00e−7
VMILA	1.69e−1	9.60e−1	23	2.52e−1	1.00e−6
satellite, SNR = 40					
ACQUIRE	7.01e−2	9.93e−1	114	1.09e+1	1.00e−5
PDAL	7.52e−2	9.91e−1	2680	2.50e+1	1.00e−7
SGP	7.00e−2	9.93e−1	356	5.72e+0	1.00e−6
SPIRAL-TV	6.62e−1	8.89e−1	61	1.03e+0	1.00e−7
SPLIT-BREGMAN	7.94e−2	9.90e−1	2114	2.50e+1	1.00e−6
VMILA	1.74e−1	9.59e−1	22	2.46e−1	1.00e−6

Table 5: Test set T2, motion blur: minimum relative error achieved by each method and corresponding MSSIM value, number of iterations, execution time and tolerance.

Test set T2, out-of-focus blur					
Method	Min rel err	MSSIM	Iters	Time	Tol
cameraman, SNR = 35					
ACQUIRE	1.18e−1	7.41e−1	12	1.11e+0	1.00e−3
PDAL	1.20e−1	7.50e−1	2624	2.50e+1	1.00e−5
SGP	1.18e−1	7.39e−1	122	1.96e+0	1.00e−5
SPIRAL-TV	1.22e−1	7.44e−1	237	2.51e+1	1.00e−5
SPLIT-BREGMAN	1.25e−1	7.49e−1	144	1.72e+0	1.00e−4
VMILA	1.19e−1	7.50e−1	89	1.36e+0	1.00e−4
cameraman, SNR = 40					
ACQUIRE	9.20e−2	7.86e−1	14	1.33e+0	1.00e−3
PDAL	9.66e−2	7.12e−1	2658	2.50e+1	1.00e−5
SGP	9.22e−2	7.81e−1	160	2.64e+0	1.00e−5
SPIRAL-TV	9.54e−2	7.92e−1	386	2.50e+1	1.00e−6
SPLIT-BREGMAN	1.10e−1	7.73e−1	1284	1.52e+1	1.00e−5
VMILA	9.20e−2	7.93e−1	148	2.33e+0	1.00e−5
satellite, SNR = 35					
ACQUIRE	1.42e−1	9.72e−1	11	1.13e+0	1.00e−2
PDAL	1.43e−1	9.70e−1	2700	2.50e+1	1.00e−5
SGP	1.43e−1	9.72e−1	70	1.07e+0	1.00e−4
SPIRAL-TV	4.39e−1	8.95e−1	63	1.07e+0	1.00e−6
SPLIT-BREGMAN	1.49e−1	9.64e−1	2101	2.50e+1	1.00e−5
VMILA	2.01e−1	9.42e−1	20	2.42e−1	1.00e−6
satellite, SNR = 40					
ACQUIRE	9.30e−2	9.87e−1	116	1.11e+1	1.00e−5
PDAL	9.91e−2	9.85e−1	2707	2.50e+1	1.00e−5
SGP	9.30e−2	9.87e−1	541	8.25e+0	1.00e−6
SPIRAL-TV	5.08e−1	8.87e−1	57	9.61e−1	1.00e−7
SPLIT-BREGMAN	1.12e−1	9.79e−1	2096	2.50e+1	1.00e−6
VMILA	2.02e−1	9.42e−1	20	2.32e−1	1.00e−6

Table 6: Test set T2, out-of-focus blur: minimum relative error achieved by each method and corresponding MSSIM value, number of iterations, execution time and tolerance.