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

Daniela di Serafino†  Germana Landi‡  Marco Viola§

July 25, 2018

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 non-negativity 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.

Key words. Image restoration, Poisson noise, TV regularization, iteratively reweighted norm, quadratic approximation.

AMS subject classifications. 90C25, 65K05, 94A08.

1 Introduction

Restoring images corrupted by Poisson noise is required in many applications, such as fluorescence microscopy [38], X-ray computed tomography (CT) [29], positron emission tomography (PET) [42], confocal microscopy [35] and astronomical imaging [43, 3]. 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 $x \in \mathbb{R}^n$ and the measured data are assumed to be a vector $y \in \mathbb{N}^m_0$, whose entries $y_j$ are samples from $m$ independent Poisson random variables $Y_j$ with probability

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

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

†Dipartimento di Matematica e Fisica, Università degli Studi della Campania “Luigi Vanvitelli”, viale A. Lincoln 5, 81100 Caserta, Italy, (daniela.diserafino@unicampania.it).

‡Dipartimento di Matematica, Università degli Studi di Bologna, Piazza di Porta S. Donato 5, 40126 Bologna, Italy (germana.landi@unibo.it).

§Dipartimento di Ingegneria Informatica, Automatica e Gestionale “Antonio Ruberti”, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy (marco.viola@uniroma1.it).
where the matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ models the observation mechanism of the imaging system and $b \in \mathbb{R}^m$, $b > 0$, models the background radiation detected by the sensors. Standard assumptions on $A$ are

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

By applying a maximum-likelihood approach [3, 42], we can estimate $x$ by minimizing the Kullback-Leibler (KL) divergence of $Ax + b$ from $y$:

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

where we set $y_j \ln(y_j/(Ax + b)_j) = 0$ 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), which has received considerable attention because of its ability of preserving edges and smoothing flat areas of the images.

Assuming, for simplicity, that $x$ 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 [12]:

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

where $X$ is supposed to satisfy some boundary conditions, e.g., periodic. This can be also written as

$$TV(x) = \sum_{i=1}^{n} ||D_1x||,$$

where

$$D_1 = \left( e_{(l-1)r+k+1}^T - e_{(l-1)r+k}^T, e_{l+r+k}^T - e_{(l-1)r+k}^T \right), \quad i = (l-1)r + k,$$

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

Thus, we are interested in solving the following problem:

$$\text{minimize} \quad D_{KL}(x) + \lambda TV(x),$$

$$\text{s.t.} \quad x \in S,$$

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

$$S = S_1 := \{ x \in \mathbb{R}^n : x \geq 0 \} \quad \text{or} \quad S = S_2 := \{ x \in \mathbb{R}^n : x \geq 0, e^T x = e^T y \},$$

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

Various approaches have been proposed to solve problem [4], mostly with $S = S_1$; 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

---

1We have implicitly assumed that $y$ has been converted into a real vector with entries ranging in the same interval as the entries of $x$. 

---

D. di Serafino, G. Landi, and M. Viola
it and using well-established techniques such as expectation-maximization methods \[30, 34\] or gradient-projection methods with suitable scaling techniques aimed at accelerating convergence \[9, 31, 49\]. 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 \[7, 28\] and the forward-backward EM method discussed in \[39\]. On the other hand, the previous methods require, at each step, the solution of a Rudin-Osher-Fatemi (ROF) denoising subproblem \[37\], which can be computed only approximately, using, e.g., the algorithms proposed in \[2, 12\]. Methods based on ADMM and SPLIT BREGMAN techniques, such as those presented in \[22, 25, 40\], do not exploit smooth TV approximations too. They generally use more memory because of auxiliary variables of the same size as \(x\) or \(y\), and require the solution of linear systems involving \(A^TA\) 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 \[1\] as a saddle-point problem and solving it by a primal-dual algorithm. In this context, an alternating extragradient scheme has been presented in \[8\], and a procedure exploiting the Chambolle-Pock algorithm \[13\] has been described in \[37\].

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, 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 faster image restorations even with low accuracy requirements on the solution of the inner problems, without affecting the quality of the reconstructed images.

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 \(b > 0\) ensure that, for any given \(y \geq 0\), \(D_{KL}\) is a nonnegative, convex, coercive, twice continuously differentiable function in \(\mathbb{R}^n_+\) (see, e.g., \([4, 22]\)). Its gradient and Hessian are given by

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

and

\[
\nabla^2 D_{KL}(x) = A^T U(x) A, \quad U(x) = \text{diag} \left( \frac{y}{Ax + b} \right),
\]

where the square root and the ratios are intended componentwise, and \(\text{diag}(v)\) denotes the diagonal matrix with diagonal entries equal to the entries of \(v\). It can be proved that \(\nabla D_{KL}\) is Lipschitz continuous \[28\], furthermore, it follows from \([6]\) that \(\nabla^2 D_{KL}\) is positive definite, i.e., \(D_{KL}\) is strictly convex, whenever \(y > 0\) and \(A\) has nullspace \(N(A) = \{0\}\). In this case, if \(x\) is constrained to be in a bounded subset of the nonnegative orthant, e.g., the set \(S_2\) in \([5]\), the minimum eigenvalue of \(\nabla^2 D_{KL}(x)\) is bounded below independently of \(x\), 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 \((\ref{eq:TV})\) admits a solution, which is unique if \(y > 0\) and \(\mathcal{N}(A) = \{0\}\). Since TV is not differentiable, we use a regularized version of it, \(TV_\mu\). Taking into account the discussion in \([46]\) about smoothed versions of TV, we consider

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

where \(\phi_\mu\) is the Huber-like function

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

It is easy to verify that \(TV_\mu\) is Lipschitz continuously differentiable and its gradient reads as follows:

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

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

\[
\nabla^2 TV_\mu(x) = \sum_{i=1}^{n} \nabla^2 \phi_\mu(||D_i x||), \\
\nabla^2 \phi_\mu(||D_i x||) = \begin{cases} 
    \frac{D_i^T D_i x}{||D_i x||} - \frac{(D_i^T D_i x)(D_i^T D_i x)^T}{||D_i x||^3} & \text{if } ||D_i x|| > \mu, \\
    \frac{D_i^T D_i x}{\mu} & \text{otherwise}.
\end{cases}
\]

Now we recall basic notions about the projected gradient. Let \(S\) be a nonempty, closed and convex set. For any continuously differentiable function \(f : D \subseteq \mathbb{R}^n \to \mathbb{R}\), with \(D\) open set containing \(S\), the projected gradient of \(f\) at \(x \in S\) is defined as the orthogonal projection of \(-\nabla f\) onto the tangent cone to \(S\) at \(x\), denoted by \(T_S(x)\):

\[
\nabla_S f(x) = \arg \min \{ ||v + \nabla f(x)|| \text{ s.t. } v \in T_S(x) \},
\]

When \(S\) is the set \(S_1\) defined in \((\ref{eq:S1})\), the tangent cone takes the form

\[
T_S(x) = \{ v \in \mathbb{R}^n : v_i \geq 0 \text{ if } x_i = 0 \}
\]

and the computation of \(\nabla_S f(x)\) is straightforward; when \(S\) is the set \(S_2\) in \((\ref{eq:S2})\),

\[
T_S(x) = \{ v \in \mathbb{R}^n : v^T v = 0 \text{ and } v_i \geq 0 \text{ if } x_i = 0 \},
\]

and \(\nabla_S f(x)\) can be efficiently determined too, thanks to the availability of low-cost algorithms for computing the projection in this case (see, e.g., \([11, 14, 15]\)).

Since the projection onto \(S\) is nonexpansive, for all \(x, \bar{x} \in S\) it is

\[
||\nabla_S f(x) - \nabla_S f(\bar{x})|| \leq ||x - \bar{x}||;
\]

furthermore,

\[
-\nabla f(x) = \nabla_S f(x) + P_{N_S(x)}(-\nabla f(x)),
\]

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

\[
N_S(x) = \{ v \in \mathbb{R}^n : v^T x \leq 0 \text{ for all } x \in S(x) \},
\]

which is the polar cone of \(T_S(x)\) (see, e.g., \([30]\) Lemma 2.2]).

Finally, it is well known that any constrained stationary point \(x^*\) of \(f\) in \(S\) is characterized by \(\nabla_S f(x^*) = 0\) and that \(||\nabla_S f||\) is lower semicontinuous on \(S\) (see, e.g., \([10]\)).
3 IRN-based inexact minimization method

We propose an iterative method for solving the problem

\[
\begin{align*}
\text{minimize} & \quad D_{KL}(x) + \lambda TV_\mu(x), \\
\text{s.t.} & \quad x \in S, \tag{9}
\end{align*}
\]

where \( S \) can be any nonempty, closed and convex subset of \( \mathbb{R}^n_+ \), although our practical interest is for the feasible sets in \( [5] \). 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 \( x^{(k)} \in S \), we consider the following quadratic approximation of \( D_{KL} \):

\[
D_{KL}(x) \approx D_{KL}^{(k)}(x) = D_{KL}(x^{(k)}) + (x - x^{(k)})^T \nabla D_{KL}(x^{(k)}) + \frac{1}{2} (x - x^{(k)})^T (\nabla^2 D_{KL}(x^{(k)}) + \gamma I)(x - x^{(k)}), \tag{10}
\]

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 \( y > 0 \), \( \mathcal{N}(A) = \{0\} \) and \( S \) is bounded.

In order to build a quadratic model of \( TV_\mu \), we use the IRN approach proposed in [36]:

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

where

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

Trivially,

\[
TV_\mu^{(k)}(x^{(k)}) = TV_\mu(x^{(k)}), \quad \nabla TV_\mu^{(k)}(x^{(k)}) = \nabla TV_\mu(x^{(k)});
\]

furthermore,

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

and hence, for any \( x \) such that \( \| D_i x^{(k)} \| \neq \mu \), the Hessian \( \nabla^2 TV_\mu^{(k)}(x^{(k)}) \) can be regarded as an approximation of \( \nabla^2 TV_\mu(x^{(k)}) \), obtained by neglecting the higher order term in the right-hand side of \( \{7\} \), 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 [44].

In the following, to simplify the notation we set

\[
F(x) = D_{KL}(x) + \lambda TV_\mu(x), \quad F_k(x) = D_{KL}^{(k)}(x) + \lambda TV_\mu^{(k)}(x).
\]

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

\[
\begin{align*}
\text{minimize} & \quad F_k(x), \\
\text{s.t.} & \quad x \in S; \tag{11}
\end{align*}
\]

and performs a line search along the direction

\[
d^{(k)} = \hat{x}^{(k)} - x^{(k)},
\]
Algorithm 1 – ACQUIRE (Algorithm based on Consecutive QUadratic and Iterative REweighted norm approximations)

1: choose $x_0 \in S$, $\eta \in (0, 1)$, $\delta \in (0, 1)$, $\{\varepsilon_k\}$ such that $\varepsilon_k > 0$ and $\lim_{k \to \infty} \varepsilon_k = 0$
2: for $k = 1, 2, \ldots$ do
3: compute an approximate solution $\hat{x}^{(k)} \in S$ to the quadratic problem (11), such that
   \[ \|\hat{x}^{(k)} - x^{(k)}\| \leq \varepsilon_k \] and $F_k(\hat{x}^{(k)}) \leq F_k(x^{(k)})$ (12)
4: $\alpha_k := 1$
5: $d^{(k)} := \hat{x}^{(k)} - x^{(k)}$
6: $x^{(k)} := x^{(k)} + \alpha_k d^{(k)}$
7: while $F(x^{(k)}) > F(x^{(k)}) + \eta \alpha_k \nabla F(x^{(k)})^T d^{(k)}$ do
8: $\alpha_k := \delta \alpha_k$
9: $x^{(k)} := x^{(k)} + \alpha_k d^{(k)}$
10: end while
11: $x^{(k+1)} := x^{(k)}$
12: end for

until an Armijo condition is satisfied, to obtain an approximation $x^{(k+1)}$ to the solution of problem (9).

This procedure is sketched in Algorithm 1 and is called ACQUIRE, which comes from “Algorithm based on Consecutive QUadratic and Iterative REweighted norm approximations”.

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 (11), but only the computation of an approximate solution such that condition (12) at line 3 of the algorithm holds, with $\lim_{k \to \infty} \varepsilon_k = 0$.

In Section 4 we also show that the first condition in (12) is satisfied if
\[ \|\nabla F_k(\hat{x}^{(k)})\| \leq \theta \|\nabla F_k(x^{(0)})\|, \] and $\theta \in (0, 1)$. Therefore, the first condition in (12) 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 $S$ is one of the sets in [5].

The second condition in (12) 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 [9, 20, 32], are suited to the solution of the inner problems (11). 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 projection 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 [5, Lemma A24].

Lemma 1 (Descent lemma) Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and let $x, y \in \mathbb{R}^n$. If there exists $L > 0$ such that
\[ \|\nabla f(x + ty) - \nabla f(x)\| \leq Lt\|y\| \text{ for all } t \in [0, 1], \]
then
\[ f(x + y) \leq f(x) + \nabla f(x)^T y + \frac{L}{2}\|y\|^2. \]
We also observe that, at step 3 of Algorithm 1 we can find \( \hat{x}^{(k)} \neq x^{(k)} \) unless \( x^{(k)} \) is the solution \( x^{(k)} \) of problem (11). However, in this case \( x^{(k)} \) is the solution of problem (7), since the gradients, and hence the projected gradients, of the objective functions of the two problems coincide at \( x^{(k)} \). Therefore, in the following we can assume that \( \hat{x}^{(k)} \neq x^{(k)} \).

The next theorem shows that the steplength \( \alpha_k \) required to obtain the iterate \( x^{(k+1)} \) 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 \( \alpha_k = \delta j_k \)

\[
F(x^{(k)}) \leq F(x^{(k)}) + \eta \alpha_k \nabla F(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}),
\]

\[
\alpha_k \geq \bar{\alpha}.
\]

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

\[
F(x^{(k)}) \leq F(x^{(k)}) + \alpha_k \nabla F(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}) + \alpha_k^2 \frac{L}{2} ||\hat{x}^{(k)} - x^{(k)}||^2,
\]

where \( L \) is the Lipschitz constant of \( \nabla F \). Then, (14) holds if we find \( \alpha_k \) such that

\[
\nabla F(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}) + \alpha_k \frac{L}{2} ||\hat{x}^{(k)} - x^{(k)}||^2 \leq \eta \nabla F(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}),
\]

or, equivalently,

\[
(1 - \eta)\nabla F(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}) + \alpha_k \frac{L}{2} ||\hat{x}^{(k)} - x^{(k)}||^2 \leq 0.
\]

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

\[
\nabla F(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}) = \nabla F_k(x^{(k)})^T(\hat{x}^{(k)} - x^{(k)}) \leq F_k(\hat{x}^{(k)}) - F_k(x^{(k)}) - \frac{\gamma}{2} ||\hat{x}^{(k)} - x^{(k)}||^2\]

\[
\leq -\frac{\gamma}{2} ||\hat{x}^{(k)} - x^{(k)}||^2,
\]

where \( \gamma \) is the strong convexity parameter of \( F_k \). Thus, (16) holds for any \( \alpha_k \) such that

\[
\frac{\gamma}{2} (\eta - 1) ||\hat{x}^{(k)} - x^{(k)}||^2 + \alpha_k \frac{L}{2} ||\hat{x}^{(k)} - x^{(k)}||^2 \leq 0.
\]

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

\[
\bar{\alpha} = \min \left\{ 1, \frac{\delta \gamma (1 - \eta)}{L} \right\}
\]

we get the thesis. \( \square \)

Now we prove that the sequence generated by ACQUIRE has a subsequence converging to a solution of problem (9). 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 \( \{x^{(k)}\} \) be the sequence generated by Algorithm 1. Then there exists a subsequence \( \{x^{(k_j)}\} \) such that

\[
\lim_{k_j \to \infty} x^{(k_j)} = \bar{x},
\]

where \( \bar{x} \in S \) is such that \( \nabla_S F(\bar{x}) = 0 \). Furthermore, any limit point \( \bar{x} \) of \( \{x^{(k)}\} \) is such that \( \nabla_S F(\bar{x}) = 0 \).
Proof. Let $\alpha_k = \delta^j_k$, where $j_k$ is given in Theorem 2. By (14) and (17) we have
\[
F(x^{(k+1)}) - F(x^{(k)}) \leq -\alpha_k \eta \frac{\gamma}{2} \|\hat{x}^{(k)} - x^{(k)}\|^2 \leq 0;
\]
then \{F(x^{(k)})\} is convergent, and the coercivity of $F$ implies that \{x^{(k)}\} is bounded. Since $\alpha_k \geq \alpha > 0$, we have that
\[
\lim_{k \to \infty} \|\hat{x}^{(k)} - x^{(k)}\| = 0 \quad (18)
\]
and \{x^{(k)}\} is bounded. This, together with $\|\hat{x}^{(k)} - x^{(k)}\| \leq \|\hat{x}^{(k)} - x^{(k)}\| + \|\hat{x}^{(k)} - x^{(k)}\|$ and the first inequality in (12), implies that
\[
\lim_{k \to \infty} \|\hat{x}^{(k)} - x^{(k)}\| = 0 \quad (19)
\]
and hence \{x^{(k)}\} is bounded. Passing to subsequences, we have
\[
\lim_{k_j \to \infty} x^{(k_j)} = \lim_{k_j \to \infty} \hat{x}^{(k_j)} = \bar{x} \in S. \quad (20)
\]
Since the projection onto a nonempty closed convex set is nonexpansive, we get
\[
\|\nabla_S F(\hat{x}^{(k_j)})\| = \|\nabla_S F(\hat{x}^{(k_j)}) - \nabla_S F_{k_j}(\hat{x}^{(k_j)})\| \leq \|\nabla F(\hat{x}^{(k_j)}) - \nabla F_{k_j}(\hat{x}^{(k_j)})\|,
\]
and, by using (20),
\[
\lim_{k_j \to \infty} \|\nabla_S F(\hat{x}^{(k_j)})\| = \lim_{k_j \to \infty} \|\nabla F(\hat{x}^{(k_j)}) - \nabla F_{k_j}(\hat{x}^{(k_j)})\| = 0.
\]
Then, for the lower semicontinuity of $\|\nabla_S F\|$, we have
\[
\nabla_S F(\bar{x}) = 0.
\]
If $\bar{x}$ is any limit point of \{x^{(k)}\}, then $\bar{x} \in S$ and, by exploiting (19) and passing to subsequences, we have
\[
\lim_{k_r \to \infty} x^{(k_r)} = \lim_{k_r \to \infty} \hat{x}^{(k_r)} = \bar{x} \in S. \quad (21)
\]
By reasoning as above we get
\[
\nabla_S F(\bar{x}) = 0,
\]
which concludes the proof. \qed

We note that ACQUIRE fits into the very general algorithmic framework presented in [21] and hence the previous convergence results can be derived by specializing and adapting the convergence theory of that framework, taking into account the specific properties of the functions $D_{KL}(x)$ and $TV\mu(x)$ and their quadratic approximations $D_{KL}^{(k)}(x)$ and $TV\mu^{(k)}(x)$, and the line search used. However, for the sake of simplicity 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 \{x^{(k)}\} converges to the minimizer of problem (9).

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

Proof. We follow the line of the proof of Lemma 2 in [6]. By Theorem 3 we know that there exists a limit point \( \mathbf{x} \) of \( \{ \mathbf{x}^{(k)} \} \) such that \( \nabla_{S} F(\mathbf{x}) = 0 \). Since \( F \) is strictly convex, \( \mathbf{x} \) is the optimal solution of problem [5]. We must prove that \( \{ \mathbf{x}^{(k)} \} \) converges to \( \mathbf{x} \).

From \( \alpha_{k} \leq 1 \) it follows that \( \| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \| \leq \| \tilde{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)} \| \) and, by (18),

\[
\lim_{k \to \infty} \| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \| = 0.
\]

Since \( \mathbf{x} \) is a strict minimizer, there exists \( \delta > 0 \) such that \( F(\mathbf{x}) < F(\mathbf{x}) \) for all \( \mathbf{x} \in S \) such that \( 0 < \| \mathbf{x} - \mathbf{x} \| \leq \delta \). For all \( \varepsilon \in (0, \delta) \), it follows from Theorem 3 that the set \( B = \{ \mathbf{x} \in S : \delta \leq \| \mathbf{x} - \mathbf{x} \| \leq \varepsilon \} \) 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 \( \mathbf{x} \). There exists \( k \in K \), \( k > k_{1} \), such that

\[
\| \mathbf{x}^{(k)} - \mathbf{x} \| < \varepsilon,
\]

and hence

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

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

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

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

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

Since \( \varepsilon \) is arbitrary, the thesis holds. \( \square \)

We conclude this section by showing that the stopping criterion (13) can be used to determine \( \tilde{\mathbf{x}}^{(k)} \) at step 3 of ACQUIRE.

Theorem 5 Assume that (13) holds for some \( \theta \in (0, 1) \). Then, there exists \( \{ \varepsilon_{k} \} \), with \( \varepsilon_{k} > 0 \) and \( \lim_{k \to \infty} \varepsilon_{k} = 0 \), such that (12) holds.

Proof. First we recall that \( -\nabla_{S} F_{k}(\mathbf{x}) = \nabla_{S} F_{k}(\mathbf{x}) + P_{N_{S}(\mathbf{x})}(-\nabla_{S} F_{k}(\mathbf{x})) \) (see 3). Since \( F_{k} \) is strongly convex with parameter \( \gamma \) and \( \mathbf{x}^{(k)} \) is the solution of problem (11), we have

\[
\begin{align*}
\frac{\gamma}{2} \| \tilde{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)} \|^2 \leq & \quad (\nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}) - \nabla_{S} F_{k}(\mathbf{x}^{(k)}))^{T}(\tilde{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) \\
= & \quad (\nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}))^{T}(\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^{(k)}) + P_{N_{S}(\mathbf{x}^{(k)})}(-\nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}))^{T}(\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^{(k)}) \\
+ & \quad P_{N_{S}(\mathbf{x}^{(k)})}(-\nabla_{S} F_{k}(\mathbf{x}^{(k)}))^{T}(\tilde{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}).
\end{align*}
\]

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

\[
\begin{align*}
\frac{\gamma}{2} \| \mathbf{x}^{(k)} - \mathbf{x}^{(k)} \|^2 \leq & \quad (\nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}))^{T}(\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^{(k)}) + P_{N_{S}(\mathbf{x}^{(k)})}(-\nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}))^{T}(\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^{(k)}) \\
+ & \quad P_{N_{S}(\mathbf{x}^{(k)})}(-\nabla_{S} F_{k}(\mathbf{x}^{(k)}))^{T}(\tilde{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}).
\end{align*}
\]

It follows that

\[
\| \mathbf{x}^{(k)} - \mathbf{x}^{(k)} \| \leq \frac{2}{\gamma} \| \nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}) \|;
\]

thus, by requiring that

\[
\| \nabla_{S} F_{k}(\tilde{\mathbf{x}}^{(k)}) \| \leq \theta^{k} \| \nabla_{S} F_{k}(\mathbf{x}^{(0)}) \|
\]

and setting \( \varepsilon_{k} = \theta^{k}(2/\gamma) \| \nabla_{S} F_{k}(\mathbf{x}^{(0)}) \| \), we get

\[
\| \mathbf{x}^{(k)} - \mathbf{x}^{(k)} \| \leq \varepsilon_{k}.
\]

\( \square \)
5 Numerical experiments

ACQUIRE was implemented in Matlab, using as inner solver the scaled gradient projection (SGP) method proposed in [9], widely applied in the solution of image restoration problems. In particular, the implementation of SGP provided by the SGP-dec Matlab code, available from http://www.unife.it/prin/software was exploited.

The SGP iteration applied to problem (11) reads:

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

where $z^{(0)} = x^{(k)}$, $\rho_j$ is a line-search parameter ensuring that $z^{(j+1)}$ 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 $P_{S_j} C_j^{-1}$ is the projection operator onto $S$ 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 (11) (see, e.g., [1, 15, 17, 18, 23, 24] and the references therein). In particular, SGP uses a modification of the ABB\textsubscript{min} adaptive Barzilai-Borwein steplength defined in [24], which takes into account the scaling matrix $C_j$ (see [9] for details); according to the analysis in [19], 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 SGP-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 [19] section 3.3 and $q$ was chosen equal to its default value in SGP-dec, i.e., $q = 3$. The SGP iterations were stopped according to (13). 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 (13) 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 SGP-dec.

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

The parameter $\gamma$ in [19] was set equal to $10^{-5}$. The nonmonotone line search proposed in [27] 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

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

i.e., when the relative change in the restored image went below a certain threshold. Six values of Tol were considered, Tol = $10^{-2}, 10^{-3}, \ldots, 10^{-7}$, 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 in the experiments.

Four reference images were chosen: cameraman, micro, phantom and satellite, shown in the first column of Figure 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 [43]; phantom is the famous Shepp-Logan brain phantom described in [41]; finally, the satellite image comes from the RestoreTools package [33]. Their sizes are reported in Table 1.

In order to define our test problems, each reference image was convolved with a Gaussian PSF with the variance $\sigma$ specified in Table 1, the function psfGauss from [33] was used to perform the convolution. In order to take into account the existence of some background emission, $10^{-10}$ was added to all pixels of the blurred image; obviously, the vector $b$ in $D_{KL}(x)$ was set as $b = 10^{-10} e$. 

The remaining features of natural choice. Setting this number to 10 was effective in our experiments. Defaults were used for the this value of $\theta$ was implemented at line 7 of Algorithm 1, with memory length equal to 5, $\eta$ that $\nu_j$ for the quadratic problem (11) (see, e.g., 1, 15, 17, 18, 23, 24) and the references therein). In particular, SGP uses a modification of the ABB\textsubscript{min} adaptive Barzilai-Borwein steplength defined in [24], which takes into account the scaling matrix $C_j$ (see [9] for details); according to the analysis in [19], 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 SGP-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 [19] section 3.3 and $q$ was chosen equal to its default value in SGP-dec, i.e., $q = 3$. The SGP iterations were stopped according to (13). 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 (13) 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 SGP-dec.

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

The parameter $\gamma$ in [19] was set equal to $10^{-5}$. The nonmonotone line search proposed in [27] 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

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

i.e., when the relative change in the restored image went below a certain threshold. Six values of Tol were considered, Tol = $10^{-2}, 10^{-3}, \ldots, 10^{-7}$, 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 in the experiments.

Four reference images were chosen: cameraman, micro, phantom and satellite, shown in the first column of Figure 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 [43]; phantom is the famous Shepp-Logan brain phantom described in [41]; finally, the satellite image comes from the RestoreTools package [33]. Their sizes are reported in Table 1.

In order to define our test problems, each reference image was convolved with a Gaussian PSF with the variance $\sigma$ specified in Table 1, the function psfGauss from [33] was used to perform the convolution. In order to take into account the existence of some background emission, $10^{-10}$ was added to all pixels of the blurred image; obviously, the vector $b$ in $D_{KL}(x)$ was set as $b = 10^{-10} e$. 
The resulting image was then corrupted by Poisson noise, using the function \texttt{imnoise} from the Matlab Image Processing Toolbox. Notice that, in case of Poisson noise, which affects the photon counting process, the 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, the intensities of the reference images were pre-scaled to get noisy and blurred images with Signal to Noise Ratio (SNR) equal to 35 and 40. These images are shown in Figures 2-5 (left columns).
Figure 2: Cameraman: corrupted and restored images (top: SNR = 35, bottom: SNR = 40).

Figure 3: Micro: corrupted and restored images (top: SNR = 35, bottom: SNR = 40).
Figure 4: Phantom: corrupted and restored images (top: SNR = 35, bottom: SNR = 40).

Figure 5: Satellite: corrupted and restored images (top: SNR = 35, bottom: SNR = 40).
The regularization parameter $\lambda$ was set by trial and error, taking into account the double goal of minimizing the relative error of the restored image with respect to the original image and getting visual satisfaction. Its values are reported in Table 1. The parameter $\mu$ in the smoothed version of TV was set as $\mu = 10^{-2}$. Before applying ACQUIRE, each corrupted image was scaled by division by its largest intensity; furthermore it was used as starting guess (i.e., $x(0) = y$).

ACQUIRE was compared with four state-of-the-art methods: SGP, SPIRAL-TV, SPLIT BREGMAN and VMILA. SPIRAL-TV is the proximal-gradient method presented in [28], a Matlab implementation of it is available from [http://drz.ac/code/](http://drz.ac/code/). By SPLIT BREGMAN we denote a version of the method proposed in [24], which was specialized for problem [4] [25] and implemented in the Matlab code tvdeconv available from [http://dev.ipol.im/~getreuer/code/](http://dev.ipol.im/~getreuer/code/). Finally, VMILA is the variable-metric inexact line-search proximal-gradient method described in [7], whose Matlab implementation can be found at [http://www.oasis.unimore.it/site/home/software.html](http://www.oasis.unimore.it/site/home/software.html). 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. For all methods, the same stopping criterion, maximum execution time, data scaling and starting guess as those used in ACQUIRE were considered. ACQUIRE was run with and without the flux constraint (i.e., using both feasible sets $S_1$ and $S_2$ – see [5]). 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 $S = S_1$.

Figures 6 and 7 show the relative errors and the execution times of each method, in seconds, versus the tolerances used in (22), for the problems with SNR = 35 and SNR = 40, respectively. The relative error was evaluated as $\|x^{(k)} - x^*\|/\|x^*\|$, where $x^*$ denotes the original image. 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 5 out of 8 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 achieves errors comparable with those of ACQUIRE, except for micro with SNR = 35, but its progress at each iteration is slower, and hence it 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 achieving small errors. VMILA is very efficient on both instances of the cameraman problem and on the phantom problem with SNR = 35, where it is faster than ACQUIRE and SGP, or comparable with them. However, there are three problems where VMILA makes very little progress in the first iterations, yielding 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 a light semiconvergence appears in the pictures representing the errors, especially for ACQUIRE. However, we were not able to completely remove it by increasing the regularization parameter without significantly deteriorating the visual quality of the image. The images obtained with ACQUIRE and corresponding to smallest errors are shown in Figures 2-5 (right columns).

Details concerning the restored images obtained with the various methods for their smallest relative errors are given in Tables 2 and 3. We report the smallest errors, the iterations performed to achieve them, and the corresponding execution times and tolerances. We also report the values of MSSIM, a structural similarity measure index [45] 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. All the data agree with the previous observations about the behavior of the methods; furthermore, 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.
Figure 6: Test problems with SNR = 35: relative error (left) and execution time (right) versus tolerance, for all methods.
Figure 7: Test problems with SNR = 40: relative error (left) and execution time (right) versus tolerance, for all methods.
Table 2: Test problems with SNR = 35: minimum relative error achieved by each method and corresponding MSSIM value, number of iterations, execution time and tolerance.

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, 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 (9) 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, ACQUIRE appears competitive with well-established methods for TV-based Poisson image restoration. We also note that our approach can be easily generalized to inverse problems with different fit-to-data terms and regularization functions.
<table>
<thead>
<tr>
<th>Method</th>
<th>Min Rel Err</th>
<th>MSSIM</th>
<th>Iters</th>
<th>Time</th>
<th>Tol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cameraman</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE</td>
<td>8.73e−2</td>
<td>8.42e−1</td>
<td>8</td>
<td>8.33e−1</td>
<td>1.00e−3</td>
</tr>
<tr>
<td>SGP</td>
<td>8.70e−2</td>
<td>8.42e−1</td>
<td>53</td>
<td>9.80e−1</td>
<td>1.00e−4</td>
</tr>
<tr>
<td>SPIRAL-TV</td>
<td>8.97e−2</td>
<td>8.36e−1</td>
<td>265</td>
<td>2.51e+1</td>
<td>1.00e−5</td>
</tr>
<tr>
<td>SPLIT BREGMAN</td>
<td>9.38e−2</td>
<td>8.41e−1</td>
<td>1824</td>
<td>2.50e+1</td>
<td>1.00e−7</td>
</tr>
<tr>
<td>VMILA</td>
<td>8.72e−2</td>
<td>8.42e−1</td>
<td>58</td>
<td>8.37e−1</td>
<td>1.00e−4</td>
</tr>
<tr>
<td><strong>micro</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE</td>
<td>4.37e−2</td>
<td>9.84e−1</td>
<td>208</td>
<td>7.00e+0</td>
<td>1.00e−5</td>
</tr>
<tr>
<td>SGP</td>
<td>4.38e−2</td>
<td>9.84e−1</td>
<td>407</td>
<td>2.20e+0</td>
<td>1.00e−6</td>
</tr>
<tr>
<td>SPIRAL-TV</td>
<td>5.16e−2</td>
<td>9.84e−1</td>
<td>1175</td>
<td>2.51e+1</td>
<td>1.00e−6</td>
</tr>
<tr>
<td>SPLIT BREGMAN</td>
<td>5.20e−2</td>
<td>9.86e−1</td>
<td>6817</td>
<td>2.50e+1</td>
<td>1.00e−7</td>
</tr>
<tr>
<td>VMILA</td>
<td>4.36e−2</td>
<td>9.87e−1</td>
<td>905</td>
<td>1.28e+1</td>
<td>1.00e−7</td>
</tr>
<tr>
<td><strong>phantom</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE</td>
<td>1.29e−1</td>
<td>9.85e−1</td>
<td>203</td>
<td>2.50e+1</td>
<td>1.00e−6</td>
</tr>
<tr>
<td>SGP</td>
<td>1.29e−1</td>
<td>9.85e−1</td>
<td>369</td>
<td>8.12e+0</td>
<td>1.00e−6</td>
</tr>
<tr>
<td>SPIRAL-TV</td>
<td>2.97e−1</td>
<td>9.10e−1</td>
<td>51</td>
<td>1.31e+0</td>
<td>1.00e−2</td>
</tr>
<tr>
<td>SPLIT BREGMAN</td>
<td>1.50e−1</td>
<td>9.83e−1</td>
<td>1802</td>
<td>2.50e+1</td>
<td>1.00e−6</td>
</tr>
<tr>
<td>VMILA</td>
<td>2.28e−1</td>
<td>9.51e−1</td>
<td>9</td>
<td>1.84e−1</td>
<td>1.00e−3</td>
</tr>
<tr>
<td><strong>satellite</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQUIRE</td>
<td>1.49e−1</td>
<td>9.69e−1</td>
<td>27</td>
<td>2.85e+0</td>
<td>1.00e−3</td>
</tr>
<tr>
<td>SGP</td>
<td>1.49e−1</td>
<td>9.69e−1</td>
<td>236</td>
<td>4.37e+0</td>
<td>1.00e−5</td>
</tr>
<tr>
<td>SPIRAL-TV</td>
<td>2.45e−1</td>
<td>9.16e−1</td>
<td>51</td>
<td>1.21e+0</td>
<td>1.00e−2</td>
</tr>
<tr>
<td>SPLIT BREGMAN</td>
<td>1.73e−1</td>
<td>9.52e−1</td>
<td>1821</td>
<td>2.50e+1</td>
<td>1.00e−5</td>
</tr>
<tr>
<td>VMILA</td>
<td>2.08e−1</td>
<td>9.37e−1</td>
<td>10</td>
<td>1.17e−1</td>
<td>1.00e−3</td>
</tr>
</tbody>
</table>

Table 3: Test problems with SNR = 40: minimum relative error achieved by each method and corresponding MSSIM value, number of iterations, execution time and tolerance.

References


