J Math Imaging Vis (2011) 40: 120–145
DOI 10.1007/s10851-010-0251-1

A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging

Antonin Chambolle · Thomas Pock

Published online: 21 December 2010

Abstract In this paper we study a first-order primal-dual algorithm for non-smooth convex optimization problems with known saddle-point structure. We prove convergence to a saddle-point with rate $O(1/N)$ in finite dimensions for the complete class of problems. We further show accelerations of the proposed algorithm to yield improved rates on problems with some degree of smoothness. In particular we show that we can achieve $O(1/N^2)$ convergence on problems, where the primal or the dual objective is uniformly convex, and we can show linear convergence, i.e. $O(\omega^N)$ for some $\omega \in (0,1)$, on smooth problems. The wide applicability of the proposed algorithm is demonstrated on several imaging problems such as image denoising, image deconvolution, image inpainting, motion estimation and multi-label image segmentation.

Keywords Convex optimization · Dual approaches · Total variation · Inverse problems · Image · Reconstruction

1 Introduction

Variational methods have proven to be particularly useful to solve a number of ill-posed inverse imaging problems. They can be divided into two fundamentally different classes: convex and non-convex problems. The advantage of convex problems over non-convex problems is that a global optimum can be computed, in many cases with a good precision and in a reasonable time, independent of the initialization. Hence, the quality of the solution solely depends on the accuracy of the model. On the other hand, non-convex problems have often the ability to model more precisely the process behind an image acquisition, but have the drawback that the quality of the solution is more sensitive to the initialization and the optimization algorithm.

Total variation minimization plays an important role in convex variational methods for imaging. The major advantage of the total variation is that it allows for sharp discontinuities in the solution. This is of vital interest for many imaging problems, since edges represent important features, e.g. object boundaries or motion boundaries. However, it is also well known that variational methods incorporating total variation regularization are difficult to minimize due to the non-smoothness of the total variation. The aim of this paper is therefore to provide a flexible algorithm which is particularly suitable for such non-smooth convex optimization problems in imaging.

In Sect. 2 we re-visit a primal-dual algorithm proposed by Pock, Bischof, Cremers and Chambolle in [29] for minimizing a convex relaxation of the Mumford-Shah functional. In subsequent work [12], Esser et al. studied the same algorithm in a more general framework and established connections to other known algorithms.

We show in this paper that this algorithm has a convergence rate of $O(1/N)$, more precisely, a *partial*[1] primal-dual gap decays like one over the total number of iterations. This rate equals the rates recently proven by Nesterov in [27]

A. Chambolle (✉)
CMAP, Ecole Polytechnique, CNRS, 91128 Palaiseau, France
e-mail: antonin.chambolle@cmap.polytechnique.fr

T. Pock
Institute for Computer Graphics and Vision,
Graz University of Technology, 8010 Graz, Austria
e-mail: pock@icg.tugraz.at

[1] The partial primal-dual gap is a weaker measure than the commonly used primal-dual gap but it can be applied to functions with unbounded domain.

and more recently by Nemirovski in [23] on (almost) the same class of problems as in this paper. We further show in Sect. 5 that for certain problems, the theoretical rate of convergence can be further improved. In particular we show that the proposed algorithm can be modified to yield a rate of convergence $O(1/N^2)$ for problems which have some regularity in the primal or in the dual objective and is linearly convergent ($O(\omega^N)$, $\omega < 1$) for smooth problems.

The primal-dual algorithm proposed in this paper can be easily adapted to different problems, is easy to implement and can be effectively accelerated on parallel hardware such as graphics processing units (GPUs). This is particularly appealing for imaging problems, where real-time applications play an important role. This is demonstrated in Sect. 6 on several variational problems such as deconvolution, zooming, inpainting, motion estimation and segmentation. We end the paper with a short discussion.

2 The General Problem

Let X, Y be two finite-dimensional real vector spaces[2] equipped with an inner product $\langle\cdot,\cdot\rangle$ and norm $\|\cdot\| = \langle\cdot,\cdot\rangle^{\frac{1}{2}}$. The map $K : X \to Y$ is a continuous linear operator with induced norm

$$\|K\| = \max\{\|Kx\| : x \in X \text{ with } \|x\| \leq 1\}. \tag{1}$$

The general problem we consider in this paper is the generic saddle-point problem

$$\min_{x\in X}\max_{y\in Y} \langle Kx, y\rangle + G(x) - F^*(y) \tag{2}$$

where $G : X \to [0, +\infty]$ and $F^* : Y \to [0, +\infty]$ are proper, convex, lower-semicontinuous (l.s.c.) functions, F^* being itself the convex conjugate of a convex l.s.c. function F. Let us observe that this saddle-point problem is a primal-dual formulation of the nonlinear primal problem

$$\min_{x\in X} F(Kx) + G(x), \tag{3}$$

or of the corresponding dual problem

$$\max_{y\in Y} -\left(G^*(-K^*y) + F^*(y)\right). \tag{4}$$

We refer to [32] for more details. We assume that these problems have at least one solution $(\hat{x}, \hat{y}) \in X \times Y$, which therefore satisfies

$$\begin{aligned} K\hat{x} &\in \partial F^*(\hat{y}),\\ -(K^*\hat{y}) &\in \partial G(\hat{x}), \end{aligned} \tag{5}$$

where ∂F^* and ∂G are the subgradients of the convex functions F^* and G. See again [32] for details. Throughout the paper we will assume that F and G are "simple", in the sense that their resolvent operator defined through

$$\begin{aligned} x &= (I + \tau\partial F)^{-1}(y)\\ &= \arg\min_x \left\{\frac{\|x-y\|^2}{2\tau} + F(x)\right\} \end{aligned}$$

has a closed-form representation (or can be efficiently solved up to a high precision, e.g. using a Newton method in low dimension). This is the case in many interesting problems in imaging, see Sect. 6. We recall that it is as easy to compute $(I + \tau\partial F)^{-1}$ as $(I + \tau\partial F^*)^{-1}$, as it is shown by the celebrated Moreau's identity:

$$x = (I + \tau\partial F)^{-1}(x) + \tau\left(I + \frac{1}{\tau}\partial F^*\right)^{-1}\left(\frac{x}{\tau}\right), \tag{6}$$

see for instance [32].

3 The Algorithm

The primal-dual algorithm we study in this paper is summarized in Algorithm 1. Note that the algorithm can also be written with $\bar{y}^{n+1} = y^{n+1} + \theta(y^{n+1} - y^n)$ instead of $\bar{x}^{n+1}$ and by exchanging the updates for y^{n+1} and x^{n+1}. We will focus on the special case $\theta = 1$ since in that case, it is relatively easy to get estimates on the convergence of the algorithm.[3] However, other cases are interesting, and in particular the semi-implicit classical Arrow-Hurwicz algorithm, which corresponds to $\theta = 0$, has been presented in the recent literature as an efficient approach for solving some type of imaging problems [37]. We'll see that in smoother cases, that approach seems indeed to perform very well, even if we can actually prove estimates for larger choices of θ. We will also consider later on (when F or G have some known regularity) some variants where the steps σ and τ and the parameter θ can be modified at each iteration, see Sect. 5.

3.1 Convergence Analysis for $\theta = 1$

For practical use, we introduce the partial primal-dual gap

$$\begin{aligned} \mathcal{G}_{B_1\times B_2}(x, y) = {} & \max_{y'\in B_2} \left\langle y', Kx\right\rangle - F^*(y') + G(x)\\ & - \min_{x'\in B_1} \left\langle y, Kx'\right\rangle - F^*(y) + G(x'). \end{aligned}$$

[2] In fact, in most of the paper X and Y could be general real Hilbert spaces with infinite dimension, however, in that case, the assumption that K has finite norm is very restrictive. We will emphasize when the assumption of finite dimensionality is really needed.

[3] Note that the convergence itself is not really an issue, in particular, it is shown in a very recent preprint [16] that for $\theta = 1$, the approach is an instance of the Proximal Point Algorithm, see [31] and Sect. 4 below.

Algorithm 1

- Initialization: Choose $\tau, \sigma > 0$, $\theta \in [0, 1]$, $(x^0, y^0) \in X \times Y$ and set $\bar{x}^0 = x^0$.
- Iterations ($n \geq 0$): Update $x^n, y^n, \bar{x}^n$ as follows:

$$\begin{cases} y^{n+1} = (I + \sigma \partial F^*)^{-1}(y^n + \sigma K \bar{x}^n) \\ x^{n+1} = (I + \tau \partial G)^{-1}(x^n - \tau K^* y^{n+1}) \\ \bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n) \end{cases} \tag{7}$$

Then, as soon as $B_1 \times B_2$ contains a saddle-point $(\hat{x}, \hat{y})$, defined by (2), we have

$$\begin{aligned} \mathcal{G}_{B_1 \times B_2}(x, y) \geq {} & \big(\langle \hat{y}, Kx \rangle - F^*(\hat{y}) + G(x)\big) \\ & - \big(\langle y, K\hat{x} \rangle - F^*(y) + G(\hat{x})\big) \geq 0. \end{aligned}$$

However, this is not a good measure of optimality, in particular, it might vanish for (x, y) which is not a saddle-point. What is easy to check is that if $\mathcal{G}_{B_1 \times B_2}(x, y) = 0$ and (x, y) lies in the *interior* of $B_1 \times B_2$, then it is a saddle-point. In practice, if B_1 and B_2 are "large enough", the gap will measure the optimality (thanks to point *(a)* in the following theorem), an important issue being that the constants will depend on these (large) unknown sets. See also Remark 3 below. In the general case we have the following result:

Theorem 1 *Let $L = \|K\|$ and assume problem (2) has a saddle-point $(\hat{x}, \hat{y})$. Choose $\theta = 1$, $\tau\sigma L^2 < 1$, and let $(x^n, \bar{x}^n, y^n)$ be defined by (7). Then:*

(a) *For any n, (x^n, y^n) remains bounded, indeed:*

$$\begin{aligned} & \frac{\|y^n - \hat{y}\|^2}{2\sigma} + \frac{\|x^n - \hat{x}\|^2}{2\tau} \\ & \quad \leq C\left(\frac{\|y^0 - \hat{y}\|^2}{2\sigma} + \frac{\|x^0 - \hat{x}\|^2}{2\tau}\right) \end{aligned} \tag{8}$$

where the constant $C \leq (1 - \tau\sigma L^2)^{-1}$;

(b) *If we let $x_N = (\sum_{n=1}^N x^n)/N$ and $y_N = (\sum_{n=1}^N y^n)/N$, for any bounded $B_1 \times B_2 \subset X \times Y$ the restricted gap has the following bound:*

$$\mathcal{G}_{B_1 \times B_2}(x_N, y_N) \leq \frac{D(B_1, B_2)}{N}, \tag{9}$$

where

$$D(B_1, B_2) = \sup_{(x,y) \in B_1 \times B_2} \frac{\|x - x^0\|^2}{2\tau} + \frac{\|y - y^0\|^2}{2\sigma}.$$

Moreover, the weak cluster points of (x_N, y_N) are saddle-points of (2);

(c) *There exists a saddle-point (x^*, y^*) such that $x^n \to x^*$ and $y^n \to y^*$.*

Remark 1 Points *(a)* and *(b)* would still hold in an infinite dimensional Hilbert setting, as can be checked from the proofs.

Proof Let us first write the iterations (7) in the general form

$$\begin{cases} y^{n+1} = (I + \sigma \partial F^*)^{-1}(y^n + \sigma K \bar{x}) \\ x^{n+1} = (I + \tau \partial G)^{-1}(x^n - \tau K^* \bar{y}). \end{cases} \tag{10}$$

We have

$$\begin{aligned} \partial F^*(y^{n+1}) &\ni \frac{y^n - y^{n+1}}{\sigma} + K\bar{x} \\ \partial G(x^{n+1}) &\ni \frac{x^n - x^{n+1}}{\tau} - K^* \bar{y} \end{aligned}$$

so that for any $(x, y) \in X \times Y$,

$$\begin{aligned} F^*(y) \geq {} & F^*(y^{n+1}) + \left\langle \frac{y^n - y^{n+1}}{\sigma}, y - y^{n+1} \right\rangle \\ & + \langle K\bar{x}, y - y^{n+1} \rangle \\ G(x) \geq {} & G(x^{n+1}) + \left\langle \frac{x^n - x^{n+1}}{\tau}, x - x^{n+1} \right\rangle \\ & - \langle K(x - x^{n+1}), \bar{y} \rangle. \end{aligned} \tag{11}$$

Summing both inequalities, it follows:

$$\begin{aligned} & \frac{\|y - y^n\|^2}{2\sigma} + \frac{\|x - x^n\|^2}{2\tau} \\ & \quad \geq \left[\left\langle Kx^{n+1}, y \right\rangle - F^*(y) + G(x^{n+1})\right] \\ & \qquad - \left[\left\langle Kx, y^{n+1} \right\rangle - F^*(y^{n+1}) + G(x)\right] \\ & \qquad + \frac{\|y - y^{n+1}\|^2}{2\sigma} + \frac{\|x - x^{n+1}\|^2}{2\tau} \\ & \qquad + \frac{\|y^n - y^{n+1}\|^2}{2\sigma} + \frac{\|x^n - x^{n+1}\|^2}{2\tau} \\ & \qquad + \left\langle K(x^{n+1} - \bar{x}), y^{n+1} - y \right\rangle \\ & \qquad - \left\langle K(x^{n+1} - x), y^{n+1} - \bar{y} \right\rangle. \end{aligned} \tag{12}$$

From this inequality it can be seen that the expression in the last line of (12) plays an important role in proving convergence of the algorithm.

The best choice of course would be to make the scheme fully implicit, i.e. $\bar{x}=x^{n+1}$ and $\bar{y}=y^{n+1}$, which however is not feasible, since this choice would require to solve problems beforehand which are as difficult as the original problem. It is easy to see that by the natural order of the iterates, the scheme can be easily made semi implicit by taking $\bar{x}=x^n$ and $\bar{y}=y^{n+1}$. This choice, corresponding to $\theta=0$ in Algorithm 1, yields the classical Arrow-Hurwicz algorithm [1] and has been used in [37] for total variation minimization. A proof of convergence for this choice is given in [12] but with some additional restrictions on the step-widths; see also Sect. 3.2 for a more detailed analysis of this scheme.

Another choice is to compute so-called leading points obtained from taking an extragradient step based on the current iterates [17, 23, 30].

Here, we consider Algorithm 1 with $\theta=1$. As in the semi-implicit case, we choose $\bar{y}=y^{n+1}$, while we choose $\bar{x}=2x^n-x^{n-1}$ which corresponds to a simple linear extrapolation based on the current and previous iterates. This can be seen as an approximate extragradient step. With this choice, the last line of (12) becomes

$$\begin{aligned}
&\left\langle K(x^{n+1}-\bar{x}), y^{n+1}-y\right\rangle - \left\langle K(x^{n+1}-x), y^{n+1}-\bar{y}\right\rangle\\
&\quad=\left\langle K((x^{n+1}-x^n)-(x^n-x^{n-1})), y^{n+1}-y\right\rangle\\
&\quad=\left\langle K(x^{n+1}-x^n), y^{n+1}-y\right\rangle - \left\langle K(x^n-x^{n-1}), y^n-y\right\rangle\\
&\qquad-\left\langle K(x^n-x^{n-1}), y^{n+1}-y^n\right\rangle\\
&\quad\geq\left\langle K(x^{n+1}-x^n), y^{n+1}-y\right\rangle - \left\langle K(x^n-x^{n-1}), y^n-y\right\rangle\\
&\qquad-L\|x^n-x^{n-1}\|\|y^{n+1}-y^n\|.
\end{aligned}\tag{13}$$

For any $\alpha>0$, we have that (using $2ab\leq\alpha a^2+b^2/\alpha$ for any a,b)

$$\begin{aligned}
L\|x^n-x^{n-1}\|\|y^{n+1}-y^n\| &\leq \frac{L\alpha\tau}{2\tau}\|x^n-x^{n-1}\|^2\\
&\quad+\frac{L\sigma}{2\alpha\sigma}\|y^{n+1}-y^n\|^2
\end{aligned}$$

and we choose $\alpha=\sqrt{\sigma/\tau}$, so that $L\alpha\tau=L\sigma/\alpha=\sqrt{\sigma\tau}L<1$.

Summing the last inequality together with (12) and (13), we get that for any $x\in X$ and $y\in Y$,

$$\begin{aligned}
&\frac{\|y-y^n\|^2}{2\sigma}+\frac{\|x-x^n\|^2}{2\tau}\\
&\quad\geq\left[\left\langle Kx^{n+1}, y\right\rangle - F^*(y)+G(x^{n+1})\right]\\
&\qquad-\left[\left\langle Kx, y^{n+1}\right\rangle - F^*(y^{n+1})+G(x)\right]\\
&\qquad+\frac{\|y-y^{n+1}\|^2}{2\sigma}+\frac{\|x-x^{n+1}\|^2}{2\tau}\\
&\qquad+(1-\sqrt{\sigma\tau}L)\frac{\|y^n-y^{n+1}\|^2}{2\sigma}\\
&\qquad+\frac{\|x^n-x^{n+1}\|^2}{2\tau}-\sqrt{\sigma\tau}L\frac{\|x^{n-1}-x^n\|^2}{2\tau}\\
&\qquad+\left\langle K(x^{n+1}-x^n), y^{n+1}-y\right\rangle\\
&\qquad-\left\langle K(x^n-x^{n-1}), y^n-y\right\rangle.
\end{aligned}\tag{14}$$

Let us now sum (14) from $n=0$ to $N-1$. It follows that for any x and y,

$$\begin{aligned}
&\sum_{n=1}^{N}\left[\langle Kx^n, y\rangle - F^*(y)+G(x^n)\right]\\
&\quad-\left[\langle Kx, y^n\rangle - F^*(y^n)+G(x)\right]\\
&\quad+\frac{\|y-y^N\|^2}{2\sigma}+\frac{\|x-x^N\|^2}{2\tau}\\
&\quad+(1-\sqrt{\sigma\tau}L)\sum_{n=1}^{N}\frac{\|y^n-y^{n-1}\|^2}{2\sigma}\\
&\quad+(1-\sqrt{\sigma\tau}L)\sum_{n=1}^{N-1}\frac{\|x^n-x^{n-1}\|^2}{2\tau}+\frac{\|x^N-x^{N-1}\|^2}{2\tau}\\
&\leq\frac{\|y-y^0\|^2}{2\sigma}+\frac{\|x-x^0\|^2}{2\tau}-\left\langle K(x^N-x^{N-1}), y^N-y\right\rangle
\end{aligned}$$

where we have introduced $x^{-1}=x^0$, in coherence with the initial choice of $\bar{x}^0=x^0$. Now, as before, $|\langle K(x^N-x^{N-1}), y^N-y\rangle|\leq\|x^N-x^{N-1}\|^2/(2\tau)+(\tau\sigma L^2)\|y-y^N\|^2/(2\sigma)$, and it follows that

$$\begin{aligned}
&\sum_{n=1}^{N}\left(\left[\langle Kx^n, y\rangle - F^*(y)+G(x^n)\right]\right.\\
&\quad\left.-\left[\langle Kx, y^n\rangle - F^*(y^n)+G(x)\right]\right)\\
&\quad+(1-\sigma\tau L^2)\frac{\|y-y^N\|^2}{2\sigma}+\frac{\|x-x^N\|^2}{2\tau}\\
&\quad+(1-\sqrt{\sigma\tau}L)\sum_{n=1}^{N}\frac{\|y^n-y^{n-1}\|^2}{2\sigma}\\
&\quad+(1-\sqrt{\sigma\tau}L)\sum_{n=1}^{N-1}\frac{\|x^n-x^{n-1}\|^2}{2\tau}\\
&\leq\frac{\|y-y^0\|^2}{2\sigma}+\frac{\|x-x^0\|^2}{2\tau}.
\end{aligned}\tag{15}$$

First we substitute $(x, y) = (\hat{x}, \hat{y})$ in (15) where $(\hat{x}, \hat{y})$ is a saddle-point of (2). Then, it follows from (2) that the first summation in (15) is non-negative, and point *(a)* in Theorem 1 follows. We then deduce from (15) and the convexity of G and F^* that, letting $x_N = (\sum_{n=1}^N x^n)/N$ and $y_N = (\sum_{n=1}^N y^n)/N$,

$$\begin{aligned}&\big[\langle Kx_N, y\rangle - F^*(y) + G(x_N)\big] \\ &\quad - \big[\langle Kx, y_N\rangle - F^*(y_N) + G(x)\big] \\ &\leq \frac{1}{N}\left(\frac{\|y-y^0\|^2}{2\sigma} + \frac{\|x-x^0\|^2}{2\tau}\right) \end{aligned}\tag{16}$$

for any $(x, y) \in X \times Y$, which yields (9). Consider now a cluster point (x^*, y^*) of (x_N, y_N) (which is a bounded sequence, hence compact). Being G and F^* convex and l.s.c., and it follows from (16) that

$$\begin{aligned}&\big[\langle Kx^*, y\rangle - F^*(y) + G(x^*)\big] \\ &\quad - \big[\langle Kx, y^*\rangle - F^*(y^*) + G(x)\big] \leq 0\end{aligned}$$

for any $(x, y) \in X \times Y$. This shows that (x^*, y^*) satisfies (2) and therefore is a saddle-point. We have shown point *(b)* in Theorem 1.

It remains to prove the convergence to a saddle-point of the whole sequence. Observe that this is the only part where we really need to assume that X and Y have finite dimension. Point *(a)* establishes that (x^n, y^n) is a bounded sequence, so that some subsequence (x^{n_k}, y^{n_k}) converges to some limit (x^*, y^*) (strongly, since the dimension of the space is finite). Observe that (15) implies that $\lim_n (x^n - x^{n-1}) = \lim_n (y^n - y^{n-1}) = 0$, in particular also x^{n_k-1} and y^{n_k-1} converge respectively to x^* and y^*. It follows that the limit (x^*, y^*) is a fixed point of the iterations (7), hence a saddle-point of our problem.

We can then take $(x, y) = (x^*, y^*)$ in (14), which we sum from $n = n^k$ to $N-1$, $N > n_k$. We obtain

$$\begin{aligned}&\frac{\|y^*-y^N\|^2}{2\sigma} + \frac{\|x^*-x^N\|^2}{2\tau} \\ &\quad + (1-\sqrt{\sigma\tau}L)\sum_{n=n_k+1}^{N} \frac{\|y^n-y^{n-1}\|^2}{2\sigma} \\ &\quad - \frac{\|x^{n_k}-x^{n_k-1}\|^2}{2\tau} + (1-\sqrt{\sigma\tau}L)\sum_{n=n_k}^{N-1}\frac{\|x^n-x^{n-1}\|^2}{2\tau} \\ &\quad + \frac{\|x^N-x^{N-1}\|^2}{2\tau} + \big\langle K(x^N-x^{N-1}), y^N-y^*\big\rangle \\ &\quad - \big\langle K(x^{n_k}-x^{n_k-1}), y^{n_k}-y^*\big\rangle \\ &\leq \frac{\|y^*-y^{n_k}\|^2}{2\sigma} + \frac{\|x^*-x^{n_k}\|^2}{2\tau}\end{aligned}$$

from which we easily deduce that $x^N \to x^*$ and $y^N \to y^*$ as $N \to \infty$. □

Remark 2 Note that when using $\tau\sigma L^2 = 1$ in (15), the control of the estimate for y^N is lost. However, one still has an estimate on x^N

$$\frac{\|x-x^N\|^2}{2\tau} \leq \frac{\|y-y^0\|^2}{2\sigma} + \frac{\|x-x^0\|^2}{2\tau}.$$

An analog estimate can be obtained by writing the algorithm in $\bar{y}$.

$$\frac{\|y-y^N\|^2}{2\sigma} \leq \frac{\|y-y^0\|^2}{2\sigma} + \frac{\|x-x^0\|^2}{2\tau}.$$

Remark 3 Let us observe that also the global gap converges with the same rate $O(1/N)$, under the additional assumption that F and G^* have full domain. More precisely, we observe that if $F^*(y)/|y| \to \infty$ as $|y| \to \infty$, then for any $R > 0$, $F^*(y) \geq R|y|$ for y large enough which yields that $\operatorname{dom} F \supset B(0, R)$. Hence F has full domain. Conversely, if F has full domain, one can check that $\lim_{|y|\to\infty} F^*(y)/|y| = +\infty$. It is classical that in this case, F is locally Lipschitz in Y. One checks, then, that

$$\max_{y\in Y}\langle y, Kx_N\rangle - F^*(y) + G(x_N) = F(Kx_N) + G(x_N)$$

is reached at some $y \in \partial F(Kx_N)$, which is globally bounded thanks to (8). It follows from (9) that $F(Kx_N) + G(x_N) - (F(K\bar{x}) + G(\bar{x})) \leq D/N$ for some constant depending on the starting point (x^0, y^0), F and L. In the same way, if $\lim_{|x|\to\infty} G(x)/|x| \to \infty$ (G^* has full domain), we have $F^*(y_N) + G^*(-K^*y_N) - (F^*(\hat{y}) + G^*(-K^*\hat{y})) \leq D/N$. If both $F^*(y)/|y|$ and $G(x)/|x|$ diverge as $|y|$ and $|x|$ go to infinity, then the global gap $\mathcal{G}(x_N, y_N) \leq D/N$.

3.2 The Arrow-Hurwicz Method ($\theta = 0$)

We have seen that the classical Arrow-Hurwicz method [1] corresponds to the choice $\theta = 0$ in Algorithm 1, that is, the particular choice $\bar{y} = y^{n+1}$ and $\bar{x} = x^n$ in (10). This leads to

$$\begin{cases} y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma Kx^n) \\ x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* y^{n+1}). \end{cases}\tag{17}$$

In [37], Zhu and Chan used this classical Arrow-Hurwicz method to solve the Rudin Osher and Fatemi (ROF) image denoising problem [33]. See also [12] for a proof of convergence of the Arrow-Hurwicz method with very small steps. A characteristic of the ROF problem (and also many others) is that the domain of F^* is bounded, i.e. $F^*(y) < +\infty \Rightarrow \|y\| \leq D$. With this assumption, we can modify the proof of Theorem 1 to show the convergence of the Arrow-Hurwicz algorithm within $O(1/\sqrt{N})$. A similar result can be found in [22]. It seems, experimentally, that the convergence is also ensured without this assumption, at least when G is uniformly convex (which is the case in [37]), see Sect. 5 and the experiments in Sect. 6.

Choosing $\bar{x} = x^n$, $\bar{y} = y^{n+1}$ in (12), we find that for any $\beta \in (0, 1]$:

$$\begin{aligned}\Big\langle K(x^{n+1} - \bar{x}), y^{n+1} - y\Big\rangle - \Big\langle K(x^{n+1} - x), y^{n+1} - \bar{y}\Big\rangle \\ = \Big\langle K(x^{n+1} - x^n), y^{n+1} - y\Big\rangle \\ \geq -\beta\frac{\|x^{n+1} - x^n\|^2}{2\tau} - \tau L^2\frac{\|y^{n+1} - y\|^2}{2\beta} \\ \geq -\beta\frac{\|x^{n+1} - x^n\|^2}{2\tau} - \tau\frac{L^2D^2}{2\beta} \end{aligned} \tag{18}$$

where $D = \operatorname{diam}(\operatorname{dom} F^*)$ and provided $F^*(y) < +\infty$. Then:

$$\begin{aligned}\sum_{n=1}^{N}\big[\langle Kx^n, y\rangle - F^*(y) + G(x^n)\big] \\ - \big[\langle Kx, y^n\rangle - F^*(y^n) + G(x)\big] \\ + \frac{\|y - y^N\|^2}{2\sigma} + \frac{\|x - x^N\|^2}{2\tau} + \sum_{n=1}^{N}\frac{\|y^n - y^{n-1}\|^2}{2\sigma} \\ + (1-\beta)\sum_{n=1}^{N}\frac{\|x^n - x^{n-1}\|^2}{2\tau} \\ \leq \frac{\|y - y^0\|^2}{2\sigma} + \frac{\|x - x^0\|^2}{2\tau} + N\tau\frac{L^2D^2}{2\beta}\end{aligned} \tag{19}$$

so that (16) is transformed in

$$\begin{aligned}\big[\langle Kx_N, y\rangle - F^*(y) + G(x_N)\big] \\ - \big[\langle Kx, y_N\rangle - F^*(y_N) + G(x)\big] \\ \leq \frac{1}{N}\left(\frac{\|y - y^0\|^2}{2\sigma} + \frac{\|x - x^0\|^2}{2\tau}\right) + \tau\frac{L^2D^2}{2\beta}.\end{aligned} \tag{20}$$

This estimate differs from our estimate (16) by an additional term, which shows that $O(1/N)$ convergence can only be guaranteed within a certain error range. Observe that by choosing $\tau = 1/\sqrt{N}$ one obtains global $O(1/\sqrt{N})$ rate of convergence of the gap, which equals the worst case rate of black-box subgradient methods [24]. In case of the ROF model, Zhu and Chan [12] showed that by a clever adaption of the step sizes the Arrow-Hurwicz method achieves much faster convergence, although a theoretical justification of the acceleration is still missing. In Sect. 5, we prove that a similar strategy applied to our algorithm also drastically improves the convergence, in case one function has some regularity property. We then have checked experimentally that our same rules, applied to the Arrow-Hurwicz method, apparently yield a similar acceleration, but a proof is still missing, see Remarks 5 and 6.

4 Connections to Existing Algorithms

In this section we establish connections to well known methods. We first establish similarities with two algorithms which are based on extrapolational gradients [17, 30]. We further show that for K being the identity, the proposed algorithm reduces to the Douglas Rachford splitting algorithm [19]. As observed in [12], we finally show that it can also be understood as a preconditioned version of the alternating direction method of multipliers.

4.1 Extrapolational Gradient Methods

We have already mentioned that the proposed algorithm shares similarities with two old methods [17, 30]. Let us briefly recall these methods to point out some connections. In order to describe these methods, it is convenient to introduce the primal-dual pair $z = \binom{x}{y}$, the convex, l.s.c. function $H(z) = G(x) + F^*(y)$, and the linear map $\bar{K} = (-K^*, K) : (Y \times X) \to (X \times Y)$.

The modified Arrow-Hurwicz method proposed by Popov in [30] can be written as

$$\begin{cases} z^{n+1} = (I + \tau\partial H)^{-1}(z^n + \tau\bar{K}\bar{z}^n), \\ \bar{z}^{n+1} = (I + \tau\partial H)^{-1}(z^{n+1} + \tau\bar{K}\bar{z}^n), \end{cases} \tag{21}$$

where $\tau > 0$ denotes the step size. This algorithm is known to converge as long as $\tau < (3L)^{-1}$, $L = \|\bar{K}\|$. Observe that in contrast to the proposed algorithm (7), Popov's algorithm requires a sequence $\{\bar{z}^n\}$ of primal and dual leading points. It therefore has a larger memory footprint and it is less efficient in cases, where the evaluation of the resolvent operators is complex.

A similar algorithm, the so-called extragradient method, had been proposed (earlier) by Korpelevich in [17]. It consists in solving:

$$\begin{cases} z^{n+1} = (I + \tau\partial H)^{-1}(z^n + \tau\bar{K}\bar{z}^n) \\ \bar{z}^{n+1} = (I + \tau\partial H)^{-1}(z^{n+1} + \tau\bar{K}z^{n+1}), \end{cases} \tag{22}$$

where $\tau < (\sqrt{2}L)^{-1}$, $L = \|\bar{K}\|$. The extragradient method bears a lot of similarities with (21), although it is not completely equivalent. In contrast to (21), the primal-dual leading point $\bar{z}^{n+1}$ is now computed by taking an extragradient step based on the current iterate. In [23], Nemirovski showed that the extragradient method converges with a rate of $O(1/N)$ for the gap.

4.2 The Douglas-Rachford Splitting Algorithm

Computing the solution of a convex optimization problem is equivalent to the problem of finding zeros of a maximal monotone operator T associated with the subgradient of the optimization problem. The proximal point algorithm [31] is probably the most fundamental algorithm for finding zeroes of T. It is written as the recursion

$$w^{n+1} = (I + \tau^n T)^{-1}(w^n), \tag{23}$$

where $\tau^n > 0$ are the steps. Unfortunately, in most interesting cases $(I + \tau^n T)^{-1}$ is hard to evaluate and hence the practical interest in the proximal point algorithm is limited.

If the operator T can be split up into a sum of two maximal monotone operators A and B such that $T = A + B$ and $(I + \tau A)^{-1}$ and $(I + \tau B)^{-1}$ are easier to evaluate than $(I+\tau T)^{-1}$, then one can devise algorithms which only need to evaluate the resolvent operators with respect to A and B. A number of different algorithms in this scenario have been proposed. Let us focus here on the Douglas-Rachford splitting algorithm (DRS) [19], which is known to be a special case of the proximal point algorithm (23), see [10]. The basic DRS algorithm is defined through the iterations

$$\begin{cases} w^{n+1} = (I + \tau A)^{-1}(2x^n - w^n) + w^n - x^n \\ x^{n+1} = (I + \tau B)^{-1}(w^{n+1}). \end{cases} \tag{24}$$

Let us now apply the DRS algorithm to the primal problem (3).[4] We let $A = K^* \partial F(K)$ and $B = \partial G$ and apply the DRS algorithm to A and B.

$$\begin{cases} w^{n+1} = \arg\min_v F(Kv) + \frac{1}{2\tau}\|v - (2x^n - w^n)\|^2 \\ \qquad\qquad + w^n - x^n \\ x^{n+1} = \arg\min_x G(x) + \frac{1}{2\tau}\|x - w^{n+1}\|^2. \end{cases} \tag{25}$$

By standard Fenchel-Rockafellar duality, we can derive the dual problem of the first minimization problem of (25). It is written

$$y^{n+1} = \arg\min_y F^*(y) + \frac{\tau}{2}\left\| K^* y - \frac{2x^n - w^n}{\tau} \right\|^2, \tag{26}$$

where the primal and dual solutions are related via

$$w^{n+1} = x^n - \tau K^* y^{n+1}. \tag{27}$$

Similarly, the dual of the minimization problem in the second line of (25) is

$$z^{n+1} = \arg\min_z G^*(z) + \frac{\tau}{2}\left\| z - \frac{w^{n+1}}{\tau} \right\|^2, \tag{28}$$

where the primal and the dual solutions are related via

$$x^{n+1} = w^{n+1} - \tau z^{n+1}. \tag{29}$$

Finally, by combining (29) with (26), substituting (27) into (28) and substituting (27) into (29) we arrive at

$$\begin{cases} y^{n+1} = \arg\min_y F^*(y) - \langle K^* y, x^n\rangle + \frac{\tau}{2}\|K^* y + z^n\|^2, \\ z^{n+1} = \arg\min_z G^*(z) - \langle z, x^n\rangle \\ \qquad\qquad + \frac{\tau}{2}\|K^* y^{n+1} + z\|^2, \\ x^{n+1} = x^n - \tau(K^* y^{n+1} + z^{n+1}). \end{cases} \tag{30}$$

This variant of the DRS algorithm is also known as the alternating method of multipliers (ADMM). Using Moreau's identity (6), we can further simplify (30),

$$\begin{cases} y^{n+1} = \arg\min_y F^*(y) - \langle K^* y, x^n\rangle + \frac{\tau}{2}\|K^* y + z^n\|^2 \\ x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* y^{n+1}) \\ z^{n+1} = \dfrac{x^n - x^{n+1}}{\tau} - K^* y^{n+1}. \end{cases} \tag{31}$$

We can now see that for $K = I$, the above scheme reduces to (7), meaning that in this case Algorithm 1 is equivalent to the DRS algorithm (24) as well as to the ADMM (30).

4.3 Preconditioned ADMM

In many practical problems, $G(x)$ and $F^*(y)$ are relatively easy to invert (e.g. total variation methods), but the minimization of the first step in (31) is still hard since it amounts to solve a least squares problem including the linear operator K. As recently observed in [12], a clever idea is to add an additional prox term of the form

$$\frac{1}{2}\left\langle M(y - y^n), y - y^n\right\rangle,$$

where M is a positive definite matrix, to the first step in (31). Then, by the particular choice

$$M = \frac{1}{\sigma} I - \tau K K^*, \qquad 0 < \tau\sigma < 1/L^2$$

the update of the first step in (31) reduces to

$$\begin{aligned} y^{n+1} &= \arg\min_y F^*(y) - \left\langle K^* y, x^n\right\rangle + \frac{\tau}{2}\left\| K^* y + z^n \right\|^2 \\ &\quad + \frac{1}{2}\left\langle \left(\frac{1}{\sigma} - \tau K K^*\right)(y - y^n), y - y^n \right\rangle \\ &= \arg\min_y F^*(y) - \left\langle y, K x^n\right\rangle + \frac{\tau}{2}\left\langle y, K K^* y\right\rangle \\ &\quad + \tau\left\langle y, K z^n\right\rangle \\ &\quad + \frac{1}{2\sigma}\langle y, y\rangle - \frac{\tau}{2}\left\langle y, K K^* y\right\rangle \end{aligned}$$

[4] Clearly, the DRS algorithm can also be applied to the dual problem.

$$
\begin{aligned}
&\quad - \left\langle y, \left(\frac{1}{\sigma} - \tau K K^*\right) y^n \right\rangle \\
&= \arg\min_y F^*(y) \\
&\quad + \frac{1}{2\sigma} \left\| y - \left(y^n + \sigma K \left(x^n - \tau(K^* y^n + z^n)\right)\right) \right\|^2 .
\end{aligned} \tag{32}
$$

This can be further simplified to

$$
y^{n+1} = (I + \sigma \partial F^*)^{-1} \left(y^n + \sigma K \bar{x}^n\right), \tag{33}
$$

where we have defined

$$
\begin{aligned}
\bar{x}^n &= x^n - \tau(K^* y^n + z^n) \\
&= x^n - \tau \left(K^* y^n + \frac{x^{n-1} - x^n}{\tau} - K^* y^n \right) \\
&= 2x^n - x^{n-1}.
\end{aligned} \tag{34}
$$

By the additional prox term the first step becomes explicit and hence, it can be understood as a preconditioner. Note that the preconditioned version of the ADMM is equivalent to the proposed primal-dual algorithm.

5 Acceleration

Is shown in [2, 25, 27, 28] that if G or F^* is uniformly convex (such that G^*, or respectively F, has a Lipschitz continuous gradient), $O(1/N^2)$ convergence can be guaranteed. Furthermore, in case both G and F^* are uniformly convex (equivalently, both G^* and F have Lipschitz continuous gradient), it is shown in [26] that linear convergence (i.e. $O(\omega^N)$, $\omega < 1$) can be achieved. In this section we show how we can modify our algorithm in order to accelerate the convergence in these situations, to the same rate.

5.1 The Case G or F^* Uniformly Convex

For simplicity we will only treat the case where G is uniformly convex, since by symmetry, the case where F^* is uniformly convex is completely equivalent. Let us assume the existence of $\gamma > 0$ such that for any $x \in \operatorname{dom} \partial G$,

$$
\begin{aligned}
&G(x') \geq G(x) + \left\langle p, x' - x \right\rangle + \frac{\gamma}{2} \|x - x'\|^2, \\
&\quad \forall p \in \partial G(x),\ x' \in X.
\end{aligned} \tag{35}
$$

In that case one can show that ∇G^* is $1/\gamma$-Lipschitz so that the dual problem (4) can be solved in $O(1/N^2)$ using any of the accelerated first order methods of [2, 25, 27], in the sense that the objective (in this case, the dual energy) approaches its optimal value at the rate $O(1/N^2)$, where N is the number of first order iterations. We explain now that a modification of our approach yields essentially the same rate of convergence. An alternative way to reach acceleration is to use a reinitialization, such as in [28]. However, we found this approach less efficient, see the Appendix in [9] for details.

In view of (5), it follows from (35) that for any saddle-point $(\hat{x}, \hat{y})$ and any $(x, y) \in X \times Y$.

$$
\begin{aligned}
&\left[\left\langle Kx, \hat{y}\right\rangle - F^*(\hat{y}) + G(x)\right] - \left[\left\langle K\hat{x}, y\right\rangle - F^*(y) + G(\hat{x})\right] \\
&\quad = G(x) - G(\hat{x}) + \left\langle K^*\hat{y}, x - \hat{x}\right\rangle + F^*(y) \\
&\qquad - F^*(\hat{y}) - \left\langle K\hat{x}, y - \hat{y}\right\rangle \\
&\quad \geq \frac{\gamma}{2}\|x - \hat{x}\|^2.
\end{aligned} \tag{36}
$$

Observe that in case G satisfies (35), we can also replace the second equation in (11) with the stronger inequality:

$$
\begin{aligned}
G(x) \geq G(x^{n+1}) &+ \left\langle \frac{x^n - x^{n+1}}{\tau}, x - x^{n+1} \right\rangle \\
&- \left\langle K(x - x^{n+1}), \bar{y} \right\rangle + \frac{\gamma}{2}\|x - x^{n+1}\|^2.
\end{aligned}
$$

Then, modifying (12) accordingly, choosing $(x, y) = (\hat{x}, \hat{y})$ a saddle-point and using (36), we deduce that

$$
\begin{aligned}
&\frac{\|\hat{y} - y^n\|^2}{2\sigma} + \frac{\|\hat{x} - x^n\|^2}{2\tau} \\
&\geq \gamma\|\hat{x} - x^{n+1}\|^2 + \frac{\|\hat{y} - y^{n+1}\|^2}{2\sigma} + \frac{\|\hat{x} - x^{n+1}\|^2}{2\tau} \\
&\quad + \frac{\|y^n - y^{n+1}\|^2}{2\sigma} + \frac{\|x^n - x^{n+1}\|^2}{2\tau} \\
&\quad + \left\langle K(x^{n+1} - \bar{x}), y^{n+1} - \hat{y} \right\rangle \\
&\quad - \left\langle K(x^{n+1} - \hat{x}), y^{n+1} - \bar{y} \right\rangle.
\end{aligned} \tag{37}
$$

Now, we will show that we can gain acceleration of the algorithm, provided we use dynamic steps (τ_n, σ_n) and a variable relaxation parameter $\theta_n \in [0, 1]$. We therefore consider the case where we choose in (37)

$$
\bar{x} = x^n + \theta_{n-1}(x^n - x^{n-1}), \qquad \bar{y} = y^{n+1}.
$$

We obtain, adapting (13), and introducing the dependence on n also for τ, σ,

$$
\begin{aligned}
&\frac{\|\hat{y} - y^n\|^2}{2\sigma_n} + \frac{\|\hat{x} - x^n\|^2}{2\tau_n} \\
&\geq \gamma\|\hat{x} - x^{n+1}\|^2 + \frac{\|\hat{y} - y^{n+1}\|^2}{2\sigma_n} + \frac{\|\hat{x} - x^{n+1}\|^2}{2\tau_n} \\
&\quad + \frac{\|y^n - y^{n+1}\|^2}{2\sigma_n} + \frac{\|x^n - x^{n+1}\|^2}{2\tau_n}
\end{aligned}
$$

Algorithm 2

- Initialization: Choose $\tau_0, \sigma_0 > 0$ with $\tau_0\sigma_0 L^2 \leq 1$, $(x^0, y^0) \in X \times Y$, and $\bar{x}^0 = x^0$.
- Iterations ($n \geq 0$): Update $x^n, y^n, \bar{x}^n, \theta_n, \tau_n, \sigma_n$ as follows:

$$\begin{cases} y^{n+1} = (I + \sigma_n \partial F^*)^{-1}(y^n + \sigma_n K\bar{x}^n) \\ x^{n+1} = (I + \tau_n \partial G)^{-1}(x^n - \tau_n K^* y^{n+1}) \\ \theta_n = 1/\sqrt{1+2\gamma\tau_n},\ \tau_{n+1} = \theta_n\tau_n,\ \sigma_{n+1} = \sigma_n/\theta_n \\ \bar{x}^{n+1} = x^{n+1} + \theta_n(x^{n+1} - x^n) \end{cases} \tag{38}$$

$$\begin{aligned} &+ \left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y} \right\rangle \\ &- \theta_{n-1}\left\langle K(x^n - x^{n-1}), y^n - \hat{y} \right\rangle \\ &- \theta_{n-1} L \|x^n - x^{n-1}\| \|y^{n+1} - y^n\|. \end{aligned}$$

It follows that

$$\begin{aligned} &\frac{\|\hat{y} - y^n\|^2}{\sigma_n} + \frac{\|\hat{x} - x^n\|^2}{\tau_n} \\ &\quad \geq (1 + 2\gamma\tau_n)\frac{\tau_{n+1}}{\tau_n}\frac{\|\hat{x} - x^{n+1}\|^2}{\tau_{n+1}} + \frac{\sigma_{n+1}}{\sigma_n}\frac{\|\hat{y} - y^{n+1}\|^2}{\sigma_{n+1}} \\ &\qquad + \frac{\|y^n - y^{n+1}\|^2}{\sigma_n} + \frac{\|x^n - x^{n+1}\|^2}{\tau_n} - \frac{\|y^n - y^{n+1}\|^2}{\sigma_n} \\ &\qquad - \theta_{n-1}^2 L^2 \sigma_n \tau_{n-1} \frac{\|x^n - x^{n-1}\|^2}{\tau_{n-1}} \\ &\qquad + 2\left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y} \right\rangle \\ &\qquad - 2\theta_{n-1}\left\langle K(x^n - x^{n-1}), y^n - \hat{y} \right\rangle. \end{aligned} \tag{39}$$

It is clear that we can get something interesting out of (39) provided we can choose the sequences $(\tau_n)_n$,$(\sigma_n)_n$ in such a way that

$$(1 + 2\gamma\tau_n)\frac{\tau_{n+1}}{\tau_n} = \frac{\sigma_{n+1}}{\sigma_n} > 1.$$

This motivates the following Algorithm 2, which is a variant of Algorithm 1.

Observe that it means choosing $\theta_{n-1} = \tau_n/\tau_{n-1}$ in (39). Now, $(1 + 2\gamma\tau_n)\tau_{n+1}/\tau_n = \sigma_{n+1}/\sigma_n = 1/\theta_n = \tau_n/\tau_{n+1}$, so that (39) becomes, denoting for each $n \geq 0$

$$\Delta_n = \frac{\|\hat{y} - y^n\|^2}{\sigma_n} + \frac{\|\hat{x} - x^n\|^2}{\tau_n},$$

dividing the equation by τ_n, and using $L^2\sigma_n\tau_n = L^2\sigma_0\tau_0 \leq 1$,

$$\begin{aligned} \frac{\Delta_n}{\tau_n} &\geq \frac{\Delta_{n+1}}{\tau_{n+1}} + \frac{\|x^n - x^{n+1}\|^2}{\tau_n^2} - \frac{\|x^n - x^{n-1}\|^2}{\tau_{n-1}^2} \\ &\quad + \frac{2}{\tau_n}\left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y} \right\rangle \\ &\quad - \frac{2}{\tau_{n-1}}\left\langle K(x^n - x^{n-1}), y^n - \hat{y} \right\rangle. \end{aligned} \tag{40}$$

It remains to sum this equation from $n = 0$ to $n = N - 1$, $N \geq 1$. We obtain (using $x^{-1} = x^0$)

$$\begin{aligned} \frac{\Delta_0}{\tau_0} &\geq \frac{\Delta_N}{\tau_N} + \frac{\|x^{N-1} - x^N\|^2}{\tau_{N-1}^2} \\ &\quad + \frac{2}{\tau_{N-1}}\left\langle K(x^N - x^{N-1}), y^N - \hat{y} \right\rangle \\ &\geq \frac{\Delta_N}{\tau_N} + \frac{\|x^{N-1} - x^N\|^2}{\tau_{N-1}^2} \\ &\quad - \frac{\|x^{N-1} - x^N\|^2}{\tau_{N-1}^2} - L^2\|y^N - \hat{y}\|^2 \end{aligned}$$

which eventually gives:

$$\begin{aligned} &\tau_N^2 \frac{1 - L^2\sigma_0\tau_0}{\sigma_0\tau_0}\|\hat{y} - y^N\|^2 + \|\hat{x} - x^N\|^2 \\ &\quad \leq \tau_N^2\left(\frac{\|\hat{x} - x^0\|^2}{\tau_0^2} + \frac{\|\hat{y} - y^0\|^2}{\sigma_0\tau_0}\right). \end{aligned} \tag{41}$$

In case one chooses exactly $\sigma_0\tau_0 L^2 = 1$, it boils down to:

$$\|\hat{x} - x^N\|^2 \leq \tau_N^2\left(\frac{\|\hat{x} - x^0\|^2}{\tau_0^2} + L^2\|\hat{y} - y^0\|^2\right). \tag{42}$$

Now, let us show that $\gamma\tau_N \sim N^{-1}$ for N (not too large), for any "reasonable" choice of τ_0. Here by "reasonable", we mean any (large) number which can be encoded on a standard computer. It will follow that our scheme shows an $O(1/N^2)$ convergence to the optimum for the variable x^N, which is the "best" known rate for this general class of problem [2, 25, 27].

Lemma 1 *Let $\lambda \in (1/2, 1)$ and assume $\gamma\tau_0 > \lambda$. Then after*

$$N \geq \frac{1}{\ln 2} \ln\left(\frac{\ln 2\gamma\tau_0}{\ln 2\lambda}\right) \tag{43}$$

iterations, one has $\gamma\tau_N \leq \lambda$.

Observe that in particular, if we take for instance $\gamma\tau_0 = 10^{20}$ and $\lambda = 3/4$, we find that $\gamma\tau_N \leq 3/4$ as soon as $N \geq 17$ (this estimate is far from optimal, as in this case we already have $\gamma\tau_7 \approx 0.546 < 3/4$).

Proof From (38) we see that $\tilde{\tau}_N = \gamma\tau_N$ follows the update rule

$$\tilde{\tau}_{N+1} = \frac{\tilde{\tau}_N}{\sqrt{1+2\tilde{\tau}_N}}, \tag{44}$$

in particular, letting for each $N \geq 0$ $s_N = 1/\tilde{\tau}_N$, it follows

$$\begin{aligned}\sqrt{2s_N} \leq s_{N+1} &= (s_N+1)\sqrt{1-\frac{1}{(s_N+1)^2}}\\ &\leq s_N + 1.\end{aligned} \tag{45}$$

From the left-hand side inequality it follows that $s_N \geq 2(s_0/2)^{1/2^N}$. Now, $\gamma\tau_N \leq \lambda$ if and only if $s_N \geq 1/\lambda$, which is ensured as soon as $(s_0/2)^{1/2^N} \geq 1/(2\lambda)$, and we deduce (43). □

Lemma 2 *Let $\lambda > 0$, and $N \geq 0$ with $\gamma\tau_N \leq \lambda$. Then for any $l \geq 0$,*

$$(\gamma\tau_N)^{-1} + \frac{l}{1+\lambda} \leq (\gamma\tau_{N+l})^{-1} \leq (\gamma\tau_N)^{-1} + l. \tag{46}$$

Proof The right-hand side inequality trivially follows from (45). Using $\sqrt{1-t} \geq 1-t$ for $t \in [0,1]$, we also deduce that

$$s_{N+1} \geq s_N + 1 - \frac{1}{s_N+1} = s_N + \frac{s_N}{s_N+1}.$$

Hence if $s_N \geq 1/\lambda$, $s_{N+l} \geq 1/\lambda$ for any $l \geq 0$ and we deduce easily the left-hand side of (46). □

Corollary 1 *One has $\lim_{N\to\infty} N\gamma\tau_N = 1$.*

We have shown the following result:

Theorem 2 *Choose $\tau_0 > 0$, $\sigma_0 = 1/(\tau_0 L^2)$, and let $(x^n, y^n)_{n\geq 1}$ be defined by Algorithm 2. Then for any $\varepsilon > 0$, there exists N_0 (depending on ε and $\gamma\tau_0$) such that for any $N \geq N_0$,*

$$\|\hat{x} - x^N\|^2 \leq \frac{1+\varepsilon}{N^2}\left(\frac{\|\hat{x}-x^0\|^2}{\gamma^2\tau_0^2} + \frac{L^2}{\gamma^2}\|\hat{y}-y^0\|^2\right).$$

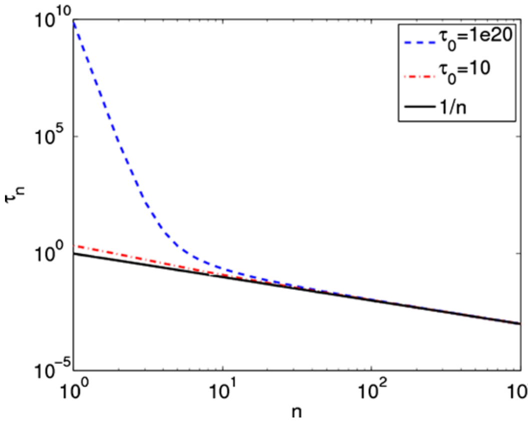


Fig. 1 The figure shows the sequence $(\gamma\tau_n)_{n\geq 1}$, using $\gamma = 1$. Observe that it goes very fast to $1/n$, in a way which is quite insensitive to the initial τ_0

Of course, the important point is that the convergence in Corollary 1 is relatively fast, for instance, if $\gamma\tau_0 = 10^{20}$, one can check that $\gamma\tau_{100} \approx 1.077/100$. Hence, the value of N_0 in Theorem 2 is never very large, even for large values of τ_0, see Fig. 1. If one has some estimate on the initial distance $\|\hat{x} - x^0\|$, a good choice is to pick $\gamma\tau_0 \gg \|\hat{x} - x^0\|$ in which case the convergence estimate boils down approximately to

$$\|\hat{x} - x^N\|^2 \lesssim \frac{1}{N^2}\left(\eta + \frac{L^2}{\gamma^2}\|\hat{y} - y^0\|^2\right)$$

with $\eta \ll 1$, and for N (not too) large enough.

Remark 4 In [2, 25, 27], the $O(1/N^2)$ estimate is theoretically better than ours since it is on the dual energy $G^*(-K^*y^N) + F^*(y^N) - (G^*(-K^*\hat{y}) + F^*(\hat{y}))$ (which can easily be shown to bound $\|x^N - \hat{x}\|^2$, see for instance [13]). In practice, however, we did not observe that our approach was slower than the other accelerated first-order schemes.

Remark 5 We have observed that replacing the last updating rule in (38) with $\bar{x}^{n+1} = x^{n+1}$, which corresponds to considering the standard Arrow-Hurwicz algorithm with varying steps as in [37], the same rate of convergence is observed (with even smaller constants, in practice) than using Algorithm 2. It seems that for an appropriate choice of the initial steps, this algorithm has also the $O(1/N^2)$ rate, however, some instabilities are observed in the convergence, see Fig. 3. On the other hand, it is relatively easy to show $O(1/N)$ convergence of this approach with the assumption that dom F^* is bounded, just as in Sect. 3.2.

5.2 The Case when G and F^* are Uniformly Convex

In case G and F^* are both uniformly convex, it is known that first order algorithms should converge linearly to the (unique) optimal value. We show that it is indeed a feature of our algorithm.

We assume that G satisfies (35), and that F^* satisfies a similar inequality with a parameter $\delta > 0$ instead of γ. In particular, (36) becomes in this case

$$\begin{aligned}&\big[\langle Kx, \hat{y}\rangle - F^*(\hat{y}) + G(x)\big] - \big[\langle Kx, y\rangle - F^*(y) + G(\hat{x})\big]\\ &\quad\geq \frac{\gamma}{2}\|x-\hat{x}\|^2 + \frac{\delta}{2}\|y-\hat{y}\|^2 \qquad (47)\end{aligned}$$

where $(\hat{x}, \hat{y})$ is the unique saddle-point of our problem. Furthermore, (11) becomes

$$\begin{aligned}F^*(y) &\geq F^*(y^{n+1}) + \left\langle \frac{y^n - y^{n+1}}{\sigma}, y - y^{n+1}\right\rangle\\ &\quad + \left\langle K\bar{x}, y - y^{n+1}\right\rangle + \frac{\delta}{2}\|y - y^{n+1}\|^2,\\ G(x) &\geq G(x^{n+1}) + \left\langle \frac{x^n - x^{n+1}}{\tau}, x - x^{n+1}\right\rangle\\ &\quad - \left\langle K(x - x^{n+1}), \bar{y}\right\rangle + \frac{\gamma}{2}\|x - x^{n+1}\|^2.\end{aligned}$$

In this case, the inequality (12), for $x = \hat{x}$ and $y = \hat{y}$, becomes

$$\begin{aligned}&\frac{\|\hat{y} - y^n\|^2}{2\sigma} + \frac{\|\hat{x} - x^n\|^2}{2\tau}\\ &\quad\geq \left(2\delta + \frac{1}{\sigma}\right)\frac{\|\hat{y} - y^{n+1}\|^2}{2} + \left(2\gamma + \frac{1}{\tau}\right)\frac{\|\hat{x} - x^{n+1}\|^2}{2}\\ &\qquad + \frac{\|y^n - y^{n+1}\|^2}{2\sigma} + \frac{\|x^n - x^{n+1}\|^2}{2\tau}\\ &\qquad + \left\langle K(x^{n+1} - \bar{x}), y^{n+1} - \hat{y}\right\rangle\\ &\qquad - \left\langle K(x^{n+1} - \hat{x}), y^{n+1} - \bar{y}\right\rangle. \qquad (48)\end{aligned}$$

Let us define $\mu = 2\sqrt{\gamma\delta}/L$, and choose σ, τ with

$$\tau = \frac{\mu}{2\gamma} = \frac{1}{L}\sqrt{\frac{\delta}{\gamma}}, \qquad \sigma = \frac{\mu}{2\delta} = \frac{1}{L}\sqrt{\frac{\gamma}{\delta}}. \qquad (49)$$

In particular we still have $\sigma\tau L^2 = 1$. Let also

$$\Delta_n := \delta\|\hat{y} - y^n\|^2 + \gamma\|\hat{x} - x^n\|^2, \qquad (50)$$

and (48) becomes

$$\begin{aligned}\Delta_n &\geq (1+\mu)\Delta_{n+1} + \delta\|y^n - y^{n+1}\|^2\\ &\quad + \gamma\|x^n - x^{n+1}\|^2 + \mu\left\langle K(x^{n+1} - \bar{x}), y^{n+1} - \hat{y}\right\rangle\\ &\quad - \mu\left\langle K(x^{n+1} - \hat{x}), y^{n+1} - \bar{y}\right\rangle. \qquad (51)\end{aligned}$$

Let us now choose

$$\bar{x} = x^n + \theta(x^n - x^{n-1}), \qquad \bar{y} = y^{n+1}, \qquad (52)$$

in (51), for $(1+\mu)^{-1} \leq \theta \leq 1$. In case $\theta = 1$ it is the same rule as in Theorem 1, but as we will see the convergence seems theoretically improved if one chooses instead $\theta = 1/(1+\mu)$. It follows from rule (52) that

$$\begin{aligned}&\left\langle K(x^{n+1} - \bar{x}), y^{n+1} - \hat{y}\right\rangle - \left\langle K(x^{n+1} - \hat{x}), y^{n+1} - \bar{y}\right\rangle\\ &\quad = \left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y}\right\rangle\\ &\qquad - \theta\left\langle K(x^n - x^{n-1}), y^{n+1} - \hat{y}\right\rangle. \qquad (53)\end{aligned}$$

We now introduce $\omega \in [(1+\mu)^{-1}, \theta]$, $\omega < 1$, which we will choose later on (if $\theta = 1/(1+\mu)$ we will obviously let $\omega = \theta$). We rewrite (53) as

$$\begin{aligned}&\left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y}\right\rangle - \omega\left\langle K(x^n - x^{n-1}), y^n - \hat{y}\right\rangle\\ &\qquad - \omega\left\langle K(x^n - x^{n-1}), y^{n+1} - y^n\right\rangle\\ &\qquad - (\theta - \omega)\left\langle K(x^n - x^{n-1}), y^{n+1} - \hat{y}\right\rangle\\ &\quad \geq \left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y}\right\rangle\\ &\qquad - \omega\left\langle K(x^n - x^{n-1}), y^n - \hat{y}\right\rangle\\ &\qquad - \omega L\left(\alpha\frac{\|x^n - x^{n-1}\|^2}{2} + \frac{\|y^{n+1} - y^n\|^2}{2\alpha}\right)\\ &\qquad - (\theta - \omega)L\left(\alpha\frac{\|x^n - x^{n-1}\|^2}{2} + \frac{\|y^{n+1} - \hat{y}\|^2}{2\alpha}\right), \qquad (54)\end{aligned}$$

for any $\alpha > 0$, and gathering (53) and (54) we obtain

$$\begin{aligned}&\mu\left\langle K(x^{n+1} - \bar{x}), y^{n+1} - \hat{y}\right\rangle\\ &\qquad - \mu\left\langle K(x^{n+1} - \hat{x}), y^{n+1} - \bar{y}\right\rangle\\ &\quad \geq \mu\Big(\left\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y}\right\rangle\\ &\qquad - \omega\left\langle K(x^n - x^{n-1}), y^n - \hat{y}\right\rangle\Big)\\ &\qquad - \mu\theta L\alpha\frac{\|x^n - x^{n-1}\|^2}{2} - \mu\omega L\frac{\|y^{n+1} - y^n\|^2}{2\alpha}\\ &\qquad - \mu(\theta - \omega)L\frac{\|y^{n+1} - \hat{y}\|^2}{2\alpha}. \qquad (55)\end{aligned}$$

From (51), (55), and choosing $\alpha = \omega(\sqrt{\gamma/\delta})$, we find

$$\Delta_n \geq \frac{1}{\omega}\Delta_{n+1} + \left(1 + \mu - \frac{1}{\omega}\right)\Delta_{n+1} + \delta\|y^n - y^{n+1}\|^2$$

Algorithm 3

- Initialization: Choose $\mu \leq 2\sqrt{\gamma\delta}/L$, $\tau = \mu/(2\gamma)$, $\sigma = \mu/(2\delta)$, and $\theta \in [1/(1+\mu), 1]$. Let $(x^0, y^0) \in X \times Y$, and $\bar{x}^0 = x^0$.
- Iterations ($n \geq 0$): Update $x^n, y^n, \bar{x}^n$ as follows:

$$\begin{cases} y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma K\bar{x}^n) \\ x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* y^{n+1}) \\ \bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n) \end{cases} \tag{56}$$

$$\begin{aligned} &+ \gamma\|x^n - x^{n+1}\|^2 + \mu\big(\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y}\rangle \\ &- \omega\langle K(x^n - x^{n-1}), y^n - \hat{y}\rangle\big) - \omega\theta\gamma\|x^{n-1} - x^n\|^2 \\ &- \delta\|y^{n+1} - y^n\|^2 - \frac{\theta - \omega}{\omega}\delta\|y^{n+1} - \hat{y}\|^2. \end{aligned} \tag{57}$$

We require that $(1 + \mu - \frac{1}{\omega}) \geq (\theta - \omega)/\omega$, which is ensured by letting

$$\omega = \frac{1+\theta}{2+\mu} = \frac{1+\theta}{2(1 + \frac{\sqrt{\gamma\delta}}{L})}. \tag{58}$$

Then, (57) becomes

$$\begin{aligned} \Delta_n \geq{}& \frac{1}{\omega}\Delta_{n+1} + \gamma\|x^n - x^{n+1}\|^2 - \omega\theta\gamma\|x^{n-1} - x^n\|^2 \\ &+ \mu\big(\langle K(x^{n+1} - x^n), y^{n+1} - \hat{y}\rangle \\ &- \omega\langle K(x^n - x^{n-1}), y^n - \hat{y}\rangle\big), \end{aligned} \tag{59}$$

which we sum from $n = 0$ to $N - 1$ after multiplying by ω^{-n}, and assuming that $x^{-1} = x^0$:

$$\begin{aligned} \Delta_0 \geq{}& \omega^{-N}\Delta_N + \omega^{-N+1}\gamma\|x^N - x^{N-1}\|^2 \\ &+ \mu\omega^{-N+1}\left\langle K(x^N - x^{N-1}), y^N - \hat{y}\right\rangle \\ \geq{}& \omega^{-N}\Delta_N + \omega^{-N+1}\gamma\|x^N - x^{N-1}\|^2 \\ &- \mu\omega^{-N+1}L\left(\sqrt{\frac{\gamma}{\delta}}\frac{\|x^N - x^{N-1}\|}{2}^2 \right. \\ &\left. + \sqrt{\frac{\delta}{\gamma}}\frac{\|y^N - \hat{y}\|}{2}^2\right) \\ \geq{}& \omega^{-N}\Delta_N - \omega^{-N+1}\delta\|y^N - \hat{y}\|^2. \end{aligned}$$

We deduce the estimate

$$\begin{aligned} &\gamma\|x^N - \hat{x}\|^2 + (1 - \omega)\delta\|y^N - \hat{y}\|^2 \\ &\quad \leq \omega^N\left(\gamma\|x^0 - \hat{x}\|^2 + \delta\|y^0 - \hat{y}\|^2\right) \end{aligned} \tag{60}$$

showing linear convergence of the iterates (x^N, y^N) to the (unique) saddle-point. We have shown the convergence of the algorithm summarized in Algorithm 3. We conclude with the following theorem.

Theorem 3 *Consider the sequence (x^n, y^n) provided by Algorithm 3 and let $(\hat{x}, \hat{y})$ be the unique solution of (2). Let $\omega < 1$ be given by (58). Then $(x^N, y^N) \to (\hat{x}, \hat{y})$ in $O(\omega^{N/2})$, more precisely, there holds (60).*

Observe that if we choose $\theta = 1$, this is an improvement over Theorem 1 (with a particular choice of the steps, given by (49)). It would be interesting to understand whether the steps can be estimated in Algorithm 1 without the a priori knowledge of γ and δ.

Remark 6 Again, we checked experimentally that taking $\theta \in [0, 1/(1+\mu)]$ in (56) also yields convergence, and sometimes faster, of Algorithm 3. In particular, the standard Arrow-Hurwicz method ($\theta = 0$) seems to work very well with these choices of τ and σ. On the other hand, it is relatively easy to show linear convergence of this method with an appropriate (different) choice of τ and σ, however, the theoretical rate is then worse than the one which we find in (58).

6 Comparisons and Applications

In this section we first present comparisons of the proposed algorithms to state-of-the-art methods. Then we illustrate the wide applicability of the proposed algorithm on several advanced imaging problems. Let us first introduce the discrete setting which we will use in the rest of this section.

6.1 Discrete Setting

We consider a regular Cartesian grid of size $M \times N$:

$$\{(ih, jh) : 1 \leq i \leq M, 1 \leq j \leq N\},$$

where h denotes the size of the spacing and (i, j) denote the indices of the discrete locations (ih, jh) in the image domain. Let $X = \mathbb{R}^{MN}$ be a finite dimensional vector space equipped with a standard scalar product

$$\langle u, v\rangle_X = \sum_{i,j} u_{i,j} v_{i,j}, \quad u, v \in X.$$

The gradient ∇u is a vector in the vector space $Y = X \times X$. For discretization of $\nabla : X \to Y$, we use standard finite differences with Neumann boundary conditions

$$(\nabla u)_{i,j} = \begin{pmatrix} (\nabla u)^1_{i,j} \\ (\nabla u)^2_{i,j} \end{pmatrix},$$

where

$$(\nabla u)^1_{i,j} = \begin{cases} \frac{u_{i+1,j}-u_{i,j}}{h} & \text{if } i < M \\ 0 & \text{if } i = M, \end{cases}$$

$$(\nabla u)^2_{i,j} = \begin{cases} \frac{u_{i,j+1}-u_{i,j}}{h} & \text{if } j < N \\ 0 & \text{if } j = N. \end{cases}$$

We also define a scalar product in Y

$$\langle p, q\rangle_Y = \sum_{i,j} p^1_{i,j} q^1_{i,j} + p^2_{i,j} q^2_{i,j},$$

$$p = (p^1, p^2), \; q = (q^1, q^2) \in Y.$$

Furthermore we will also need the discrete divergence operator $\operatorname{div} p : Y \to X$, which is chosen to be adjoint to the discrete gradient operator. In particular, one has $-\operatorname{div} = \nabla^*$ which is defined through the identity

$$\langle \nabla u, p\rangle_Y = -\langle u, \operatorname{div} p\rangle_X .$$

We also need to compute a bound on the norm of the linear operator ∇. According to (1), one has

$$L^2 = \|\nabla\|^2 = \|\operatorname{div}\|^2 \le 8/h^2.$$

See again [6] for a proof.

6.2 Total Variation Based Image Denoising

In order to evaluate and compare the performance of the proposed primal-dual algorithm to state-of-the-art methods, we will consider three different convex image denoising models, each having a different degree of regularity. Throughout the experiments, we will make use of the following procedure to determine the performance of each algorithm. We first run a well performing method for a very long time (∼100000 iterations) in order to compute a "ground truth" solution. Then, we apply each algorithm until the error (based on the solution or the energy) to the pre-determined ground truth solution is below a certain threshold ε. For each algorithm, the parameters are optimized to give an optimal performance, but stay constant for all experiments. All algorithms were implemented in Matlab and executed on a 2.66 GHz CPU, running a 64 Bit Linux system.

6.2.1 *The ROF Model*

As a prototype for total variation methods in imaging we recall the total variation based image denoising model proposed by Rudin, Osher and Fatemi in [33]. The ROF model is defined as the variational problem

$$\min_x \int_\Omega |Du| + \frac{\lambda}{2} \|u - g\|_2^2, \tag{61}$$

where $\Omega \subset \mathbb{R}^d$ is the d-dimensional image domain, $u \in L^1(\Omega)$ is the sought solution and $g \in L^1(\Omega)$ is the noisy input image. The parameter λ is used to define the tradeoff between regularization and data fitting. The term $\int_\Omega |Du|$ is the total variation of the function u, where Du denotes the distributional derivative, which is, in an integral sense, also well-defined for discontinuous functions. For sufficiently smooth functions u, e.g. $u \in W^{1,1}(\Omega)$ it reduces to $\int_\Omega |\nabla u|\, dx$. The main advantage of the total variation and hence of the ROF model is its ability to preserve sharp edges in the image, which is important for many imaging problems. Using the discrete setting introduced above (in dimension $d = 2$), the discrete ROF model, which we call the primal ROF problem is then given by

$$h^2 \min_{u\in X} \|\nabla u\|_1 + \frac{\lambda}{2}\|u - g\|_2^2, \tag{62}$$

where $u, g \in X$ are the unknown solution and the given noisy data. The norm $\|u\|_2^2 = \langle u, u\rangle_X$ denotes the standard squared L^2 norm in X and $\|\nabla u\|_1$ denotes the discrete version of the isotropic total variation norm defined as

$$\|\nabla u\|_1 = \sum_{i,j} |(\nabla u)_{i,j}|,$$

$$|(\nabla u)_{i,j}| = \sqrt{((\nabla u)^1_{i,j})^2 + ((\nabla u)^2_{i,j})^2}.$$

Casting (62) in the form of (3), we see that $F(\nabla u) = \|\nabla u\|_1$ and $G(u) = \frac{\lambda}{2}\|u - g\|_2^2$. Note that in what follows, we will always disregard the multiplicative factor h^2 appearing in the discretized energies such as (62), since it causes only a rescaling of the energy and does not change the solution.

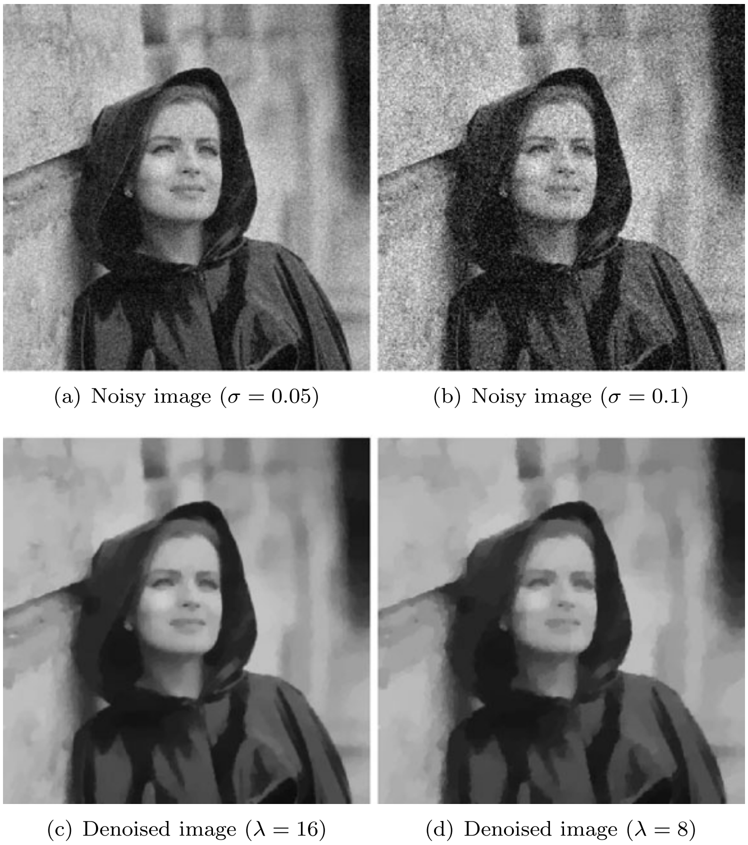
According to (2), the primal-dual formulation of the ROF problem is given by

$$\min_{u\in X} \max_{p\in Y} -\langle u, \operatorname{div} p\rangle_X + \frac{\lambda}{2}\|u - g\|_2^2 - \delta_P(p), \tag{63}$$

where $p \in Y$ is the dual variable. The convex set P is given by

$$P = \{p \in Y : \|p\|_\infty \le 1\}, \tag{64}$$

Fig. 2 Image denoising using the ROF model. The *left column* shows the noisy input image of size 256 × 256, with additive zero mean Gaussian noise ($\sigma = 0.05$) and the denoised image using $\lambda = 16$. The *right column* shows the noisy input image but now with ($\sigma = 0.1$) and the denoised image using $\lambda = 8$


(a) Noisy image ($\sigma = 0.05$)

(b) Noisy image ($\sigma = 0.1$)

(c) Denoised image ($\lambda = 16$)

(d) Denoised image ($\lambda = 8$)

and $\|p\|_\infty$ denotes the discrete maximum norm defined as

$$\|p\|_\infty = \max_{i,j} |p_{i,j}|, \qquad |p_{i,j}| = \sqrt{(p^1_{i,j})^2 + (p^2_{i,j})^2}.$$

Note that the set P is the product of pointwise L^2 balls. The function δ_P denotes the indicator function of the set P which is defined as

$$\delta_P(p) = \begin{cases} 0 & \text{if } p \in P, \\ +\infty & \text{if } p \notin P. \end{cases} \tag{65}$$

Furthermore, the primal ROF problem (62) and the primal-dual ROF problem (63) and are equivalent to the dual ROF problem

$$\max_{p \in Y} - \left(\frac{1}{2\lambda} \|\operatorname{div} p\|_2^2 + \langle g, \operatorname{div} p \rangle_X + \delta_P(p) \right). \tag{66}$$

In order to apply the proposed algorithms to (63), it remains to detail the resolvent operators $(I + \sigma \partial F^*)^{-1}$ and $(I + \tau \partial G)^{-1}$. First, casting (63) in the form of the general saddle-point problem (2) we see that $F^*(p) = \delta_P(p)$ and $G(u) = \frac{\lambda}{2}\|u - g\|_2^2$. Since F^* is the indicator function of a convex set, the resolvent operator reduces to pointwise Euclidean projectors onto L^2 balls

$$p = (I + \sigma \partial F^*)^{-1}(\tilde{p}) \quad \Longleftrightarrow \quad p_{i,j} = \frac{\tilde{p}_{i,j}}{\max(1, |\tilde{p}_{i,j}|)}.$$

The resolvent operator with respect to G poses simple pointwise quadratic problems. The solution is trivially given by

$$u = (I + \tau \partial G)^{-1}(\tilde{u}) \quad \Longleftrightarrow \quad u_{i,j} = \frac{\tilde{u}_{i,j} + \tau \lambda g_{i,j}}{1 + \tau \lambda}.$$

Observe that $G(u)$ is uniformly convex with convexity parameter λ and hence we can make use of the accelerated $O(1/N^2)$ algorithm.

Figure 2 shows the denoising capability of the ROF model using different noise levels. Note that the ROF model efficiently removes the noise while preserving the discontinuities in the image. For performance evaluation, we use the following algorithms and parameter settings:

- ALG1: $O(1/N)$ primal-dual algorithm as described in Algorithm 1, with $\tau = 0.01$, $\tau\sigma L^2 = 1$, taking the last iterate instead of the average.
- ALG2: $O(1/N^2)$ primal-dual algorithm as described in Algorithm 2, with adaptive steps, $\tau_0 = 1/L$, $\tau_n \sigma_n L^2 = 1$, $\gamma = 0.35\lambda$.
- ALG4: $O(1/N^2)$ algorithm described in the appendix of [9], based on ALG1 and reinitializations. The parameters are $q = 1$, $N_0 = 1$, $r = 2$, $\gamma = \lambda$, and we used the last iterate in the inner loop instead of the averages.
- AHMOD: Arrow-Hurwicz primal-dual algorithm (17) using the rule described in (38), $\tau_0 = 0.02$, $\tau_n \sigma_n L^2/4 = 1$, $\gamma = 0.35\lambda$.

Table 1 Performance evaluation using the images shown in Fig. 2. The entries in the table refer to the number of iterations, respectively the CPU times in seconds, which were needed to drop the root mean squared error of the solution below the error tolerance ε. The "–" entries indicate that the algorithm failed to drop the error below ε within a maximum number of 100000 iterations

	$\lambda = 16$		$\lambda = 8$	
	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$
ALG1	214 (3.38 s)	19544 (318.35 s)	309 (5.20 s)	24505 (392.73 s)
ALG2	108 (1.95 s)	937 (14.55 s)	174 (2.76 s)	1479 (23.74 s)
ALG4	124 (2.07 s)	1221 (19.42 s)	200 (3.14 s)	1890 (29.96 s)
AHMOD	64 (0.91 s)	498 (6.99 s)	122 (1.69 s)	805 (10.97 s)
AHZC	65 (0.98 s)	634 (9.19 s)	105 (1.65 s)	1001 (14.48 s)
FISTA	107 (2.11 s)	999 (20.36 s)	173 (3.84 s)	1540 (29.48 s)
NEST	106 (3.32 s)	1213 (38.23 s)	174 (5.54 s)	1963 (58.28 s)
ADMM	284 (4.91 s)	25584 (421.75 s)	414 (7.31 s)	33917 (547.35 s)
PGD	620 (9.14 s)	58804 (919.64 s)	1621 (23.25 s)	–
CFP	1396 (20.65 s)	–	3658 (54.52 s)	–

- AHZC: Arrow-Hurwicz primal-dual algorithm (17) with adaptive steps proposed by Zhu and Chan in [37].
- FISTA: $O(1/N^2)$ fast iterative shrinkage thresholding algorithm on the dual ROF problem (66) [2, 25]. We also tested a monotone variant of FISTA, called MFISTA, which however showed exactly the same performance on the tested problems.
- NEST: $O(1/N^2)$ algorithm proposed by Nesterov in [27], on the dual ROF problem (66).
- ADMM: Alternating direction method of multipliers (30), on the dual ROF problem (66), $\tau = 20$. (See also [11, 15, 19].) Two Jacobi iterations to approximately solve the linear sub-problem.
- PGD: $O(1/N)$ projected (sub)gradient descent on the dual ROF problem (66) [2, 7].
- CFP: Chambolle's fixed-point algorithm proposed in [6], on the dual ROF problem (66).

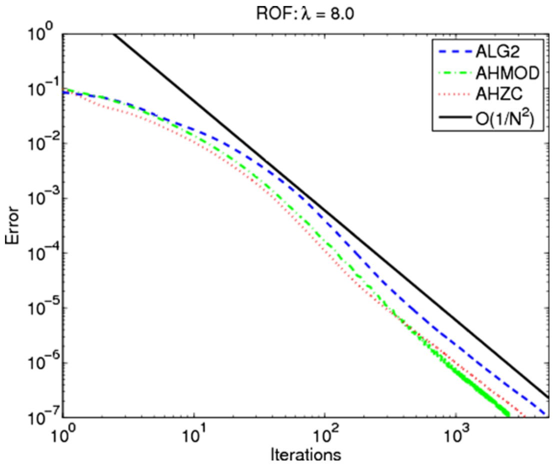
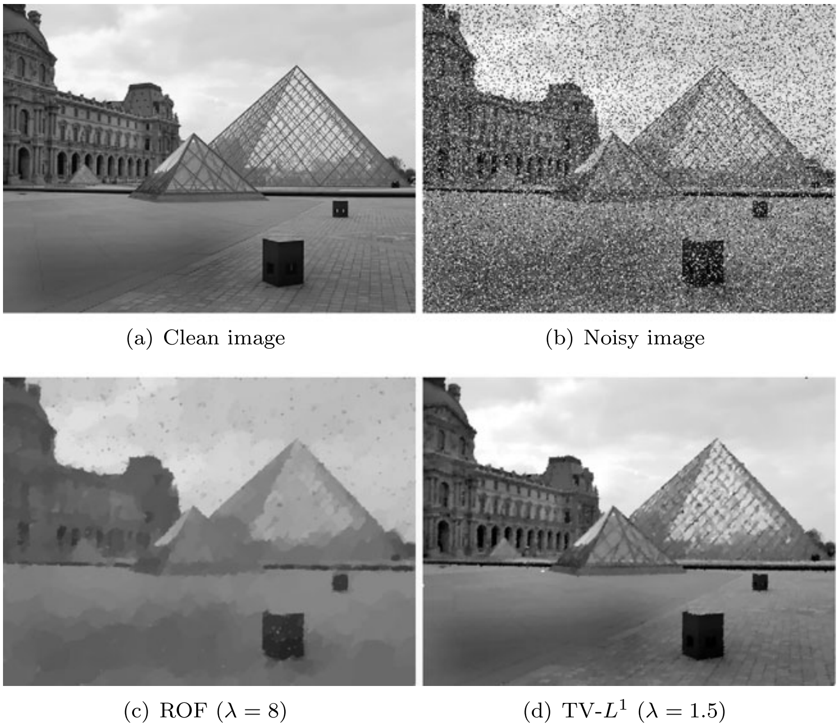


Fig. 3 Convergence of AHZC and ALG2 for the experiment in the last column of Table 1

Table 1 shows the results of the performance evaluation for the images showed in Fig. 2. One can see that the ROF problem gets harder for stronger regularization. This is explained by the fact that for stronger regularization the data term becomes less important and hence the TV-regularization term dominates. Furthermore, one can see that the theoretical efficiency rates of the algorithms are well reflected by the experiments. For the $O(1/N)$ methods, the number of iterations are increased by approximately a factor of 100 when decreasing the error threshold by a factor of 100. In contrast, for the $O(1/N^2)$ methods, the number of iterations is only increased by approximately a factor of 10. The Arrow-Hurwicz type methods (AHMOD, AHZC) appear to be the fastest algorithms. This still remains a mystery, since we do not have a theoretical explanation yet. Interestingly, by using our acceleration rule, AHMOD even outperforms AHZC. The performance of ALG2 is slightly worse, but still outperforms well established algorithms such as FISTA and NEST. Figure 3 plots the convergence of AHZC and ALG together with the theoretical $O(1/N^2)$ rate. ALG4 appears to be slightly worse than ALG2, which is also justified theoretically. ALG1 appears to be the fastest $O(1/N)$ method, but note that the $O(1/N)$ methods quickly become infeasible when requiring a higher accuracy. Interestingly, ADMM, which is often considered to be a fast method for solving L^1 related problems, seems to be slow in our experiments. PGD and CFP are competitive only, when requiring a low accuracy.

6.2.2 The TV-L^1 Model

The TV-L^1 model is obtained as a variant of the ROF model (61) by replacing the squared L^2 norm in the data



(a) Clean image

(b) Noisy image

(c) ROF ($\lambda = 8$)

(d) TV-L^1 ($\lambda = 1.5$)

Fig. 4 Image denoising in the case of impulse noise. (**a**) shows the 500 × 375 input image and (**b**) is a noisy version which has been corrupted by 25% salt and pepper noise. (**c**) is the result of the ROF model. (**d**) is the result of the TV$-L^1$ model. Note that the $TV - L^1$ model is able to remove the noise while still preserving some small details

term by the robust L^1 norm.

$$\min_u \int_\Omega |Du| + \lambda \|u - g\|_1. \tag{67}$$

Although only a slight change, the TV-L^1 model offers some potential advantages over the ROF model. First, one can check that it is contrast invariant. Second, it turns out that the TV-L^1 model is much more effective in removing noise containing strong outliers (e.g. salt&pepper noise). The discrete version of (67) is given by

$$\min_{u \in X} \|\nabla u\|_1 + \lambda \|u - g\|_1. \tag{68}$$

In analogy to (63), the saddle-point formulation of (68) is given by

$$\min_{u \in X} \max_{p \in Y} - \langle u, \operatorname{div} p \rangle_X + \lambda \|u - g\|_1 - \delta_P(p). \tag{69}$$

Comparing with the ROF problem (63), we see that the only difference is that the function $G(u)$ is now $G(u) = \lambda \|u - g\|_1$, and hence we only have to change the resolvent operator with respect to G. The solution of the resolvent operator is given by the pointwise shrinkage operations

$$u = (I + \tau \partial G)^{-1}(\tilde{u}) \quad \Longleftrightarrow$$
$$u_{i,j} = \begin{cases} \tilde{u}_{i,j} - \tau\lambda & \text{if } \tilde{u}_{i,j} - g_{i,j} > \tau\lambda \\ \tilde{u}_{i,j} + \tau\lambda & \text{if } \tilde{u}_{i,j} - g_{i,j} < -\tau\lambda \\ g_{i,j} & \text{if } |\tilde{u}_{i,j} - g_{i,j}| \le \tau\lambda. \end{cases}$$

Observe that in contrast to the ROF model, the TV-L^1 model poses a non-smooth optimization problem. Hence, we have to apply the proposed $O(1/N)$ primal-dual algorithm.

Figure 4 shows an example of outlier removal using the TV-L^1 model. Note that while the ROF leads to an over-regularized result, the TV-L^1 model efficiently removes the outliers while preserving small details. Next, we compare the performance of different algorithms on the TV-L^1 problem. For performance evaluation, we use the following algorithms and parameters:

- ALG1: $O(1/N)$ primal dual algorithm as described in Algorithm 1, $\tau = 0.02$, $\tau\sigma L^2 = 1$.
- ADMM: Alternating direction method of multipliers (30), on the dual TV-L^1 problem, $\tau = 10$ (see also [11]). Two Jacobi iterations to approximately solve the linear subproblem.
- EGRAD: $O(1/N)$ extragradient method (22), step size $\tau = 1/\sqrt{2L^2}$ (see also [17, 23]).
- NEST: $O(1/N)$ method proposed by Nesterov in [27], on the primal TV-L^1 problem, smoothing parameter $\mu = \varepsilon$.

Table 2 presents the results of the performance evaluation for the image shown in Fig. 4. Since the solution of the TV-L^1 model is in general not unique, we can not compare the RMSE of the solution. Instead we use the normalized error of the primal energy $(E^n - E^*)/E^* > 0$, where E^n is the primal energy of the current iterate n and E^* is the primal energy of the true solution, as a stopping criterion. ALG1 appears to be the fastest algorithm, followed by ADMM. Figure 5 plots the convergence of ALG1 and

Table 2 Performance evaluation using the image shown in Fig. 4. The entries in the table refer to the number of iterations, respectively the CPU times in seconds, which were needed to drop the normalized error of the primal energy below the error tolerance ε

	$\lambda = 1.5$	
	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$
ALG1	187 (15.81 s)	421 (36.02 s)
ADMM	385 (33.26 s)	916 (79.98 s)
EGRAD	2462 (371.13 s)	8736 (1360.00 s)
NEST	2406 (213.41 s)	15538 (1386.95 s)

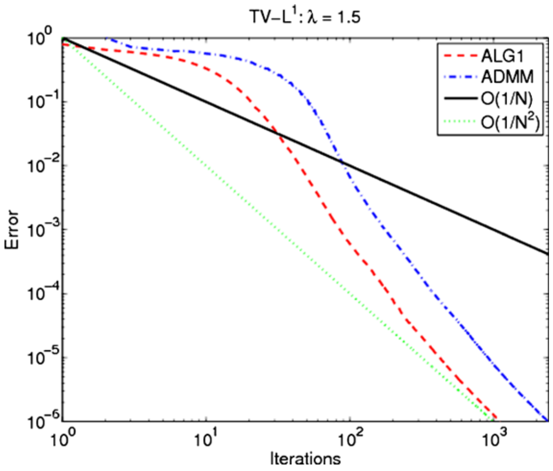


Fig. 5 Convergence for the TV-L^1 model

ADMM together with the theoretical $O(1/N)$ bound. Note that again, the proposed primal-dual algorithm significantly outperforms the state-of-the-art method ADMM. Paradoxically, it seems that both ALG1 and ADMM converge like $O(1/N^2)$ in the end, but we do not have any explanation for this yet.

6.2.3 The Huber-ROF Model

Total Variation methods applied to image regularization suffer from the so-called staircasing problem. The effect refers to the formation of artificial flat areas in the solution (see Fig. 6 (b)). A remedy for this unwanted effect is to replace the L^1 norm in the total variation term by the Huber-norm

$$|x|_\alpha = \begin{cases} \frac{|x|^2}{2\alpha} & \text{if } |x| \le \alpha \\ |x| - \frac{\alpha}{2} & \text{if } |x| > \alpha \end{cases} \tag{70}$$

where $\alpha > 0$ is a small parameter defining the tradeoff between quadratic regularization (for small values) and total variation regularization (for larger values).

This change can be easily integrated into the primal-dual ROF model (63) by replacing the term $F^*(p) = \delta_P(p)$ by $F^*(p) = \delta_P(p) + \frac{\alpha}{2}\|p\|_2^2$. Hence the primal-dual formulation of the Huber-ROF model is given by

$$\min_{u \in X} \max_{p \in Y} -\langle u, \operatorname{div} p\rangle_X + \frac{\lambda}{2}\|u-g\|_2^2 - \delta_P(p) - \frac{\alpha}{2}\|p\|_2^2. \tag{71}$$

Table 3 Performance evaluation using the image shown in Fig. 6. The entries in the table refer to the number of iterations respectively the CPU times in seconds the algorithms needed to drop the root mean squared error below the error tolerance ε

	$\lambda = 5, \alpha = 0.05$
	$\varepsilon = 10^{-15}$
ALG3	187 (3.85 s)
NEST	248 (5.52 s)

Consequently, the resolvent operator with respect to F^* is given by the pointwise operations

$$p = (I + \sigma\partial F^*)^{-1}(\tilde{p}) \iff p_{i,j} = \frac{\frac{\tilde{p}_{i,j}}{1+\sigma\alpha}}{\max(1, |\frac{\tilde{p}_{i,j}}{1+\sigma\alpha}|)}.$$

Note that the Huber-ROF model is uniformly convex in $G(u)$ and $F^*(p)$, with convexity parameters λ and α. Therefore, we can make use of the linearly convergent algorithm.

Figure 6 shows a comparison between the ROF model and the Huber-ROF model. While the ROF model leads to the development of artificial discontinuities (staircasing-effect), the Huber-ROF model yields a piecewise smooth, and hence more natural results.

For the performance evaluation, we use the following algorithms and parameter settings:

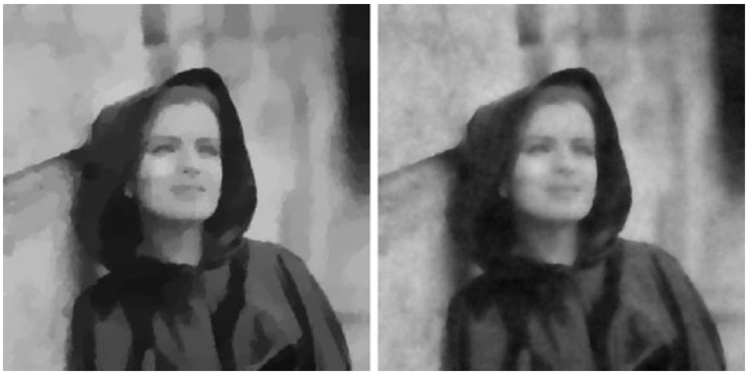
- ALG3: Linearly convergent primal-dual algorithm as described in Algorithm 3, using the convexity parameters $\gamma = \lambda$, $\delta = \alpha$, $\mu = 2\sqrt{\gamma\delta}/L$, $\theta = 1/(1+\mu)$.
- NEST: Restarted version of Nesterov's algorithm [2, 25, 28], on the dual Huber-ROF problem. Algorithm is restarted every $k = \lceil\sqrt{8L_H/\alpha}\rceil$ iterations, where $L_H = L^2/\lambda + \alpha$ is the Lipschitz constant of the dual Huber-ROF model (with our choices, $k = 17$).

Table 3 shows the result of the performance evaluation. Both ALG3 and NEST show linear convergence whereas ALG3 has a slightly better performance. Figure 7 plots the convergence of ALG3, NEST together with the theoretical $O(\omega^{N/2})$ bound. Note that ALG3 reaches machine precision after approximately 200 iterations.

6.3 Advanced Imaging Problems

In this section, we illustrate the wide applicability of the proposed primal-dual algorithms to advanced imaging problems such as image deconvolution, image inpainting, motion

Fig. 6 Comparison between the ROF model and the Huber-ROF model for the noisy image shown in Fig. 2 (b). While the ROF model exhibits strong staircasing, the Huber-ROF model leeds to piecewise smooth, and hence more natural images



(a) ROF ($\lambda = 8$) (b) Huber-ROF ($\lambda = 5$, $\alpha = 0.05$)

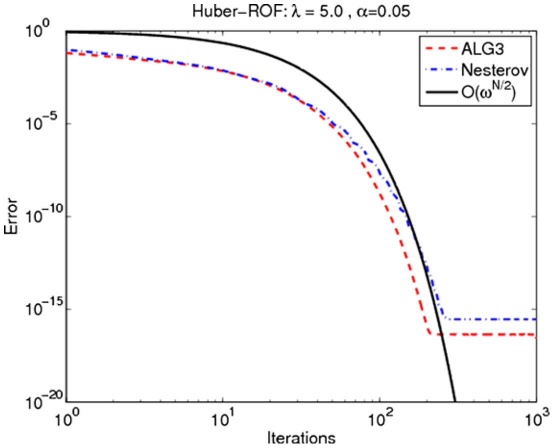


Fig. 7 Linear convergence of ALG3 and NEST for the Huber-ROF model. Note that after approximately 200 iterations, ALG3 reaches machine precision

estimation, and image segmentation. We show that the proposed algorithms can be easily adapted to all these applications and yield state-of-the-art results.

6.3.1 *Image Deconvolution and Zooming*

The standard ROF model (61) can be easily extended for image deconvolution and digital zooming.

$$\min_u \int_\Omega |Du| + \frac{\lambda}{2}\|Au - g\|_2^2, \tag{72}$$

where A is a linear operator. In the case of image deconvolution, A is the convolution with the point spread function (PSF). In the case of image zooming, A describes the downsampling procedure, which is often assumed to be a blurring kernel followed by subsampling operator. In the discrete setting, this problem can be easily rewritten in terms of a saddle-point problem

$$\min_{u\in X}\max_{p\in Y} -\langle u, \operatorname{div} p\rangle_X + \frac{\lambda}{2}\|Au - g\|_2^2 - \delta_P(p). \tag{73}$$

Now, the question is how to implement the resolvent operator with respect to $G(u) = \frac{\lambda}{2}\|Au - g\|_2^2$. In case Au can be written as a convolution, i.e. $Au = k_A * u$, where k_A is the convolution kernel, FFT based method can be used to compute the resolvent operator.

$$\begin{aligned} & u = (I + \tau\partial G)^{-1}(\tilde{u}) \\ \Longleftrightarrow \quad & u = \arg\min_u \frac{\|u - \tilde{u}\|}{2\tau} + \frac{\lambda}{2}\|k_A * u - g\|_2^2 \\ \Longleftrightarrow \quad & u = \mathcal{F}^{-1}\left(\frac{\tau\lambda\mathcal{F}(g)\mathcal{F}(k_A)^* + \mathcal{F}(\tilde{u})}{\tau\lambda\mathcal{F}(k_A)^2 + 1}\right), \end{aligned}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the FFT and inverse FFT, respectively. According to the well-known convolution theorem, the multiplication and division operators are understood pointwise in the above formula. Note that only one FFT and one inverse FFT are required to evaluate the resolvent operator (all other quantities can be pre-computed).

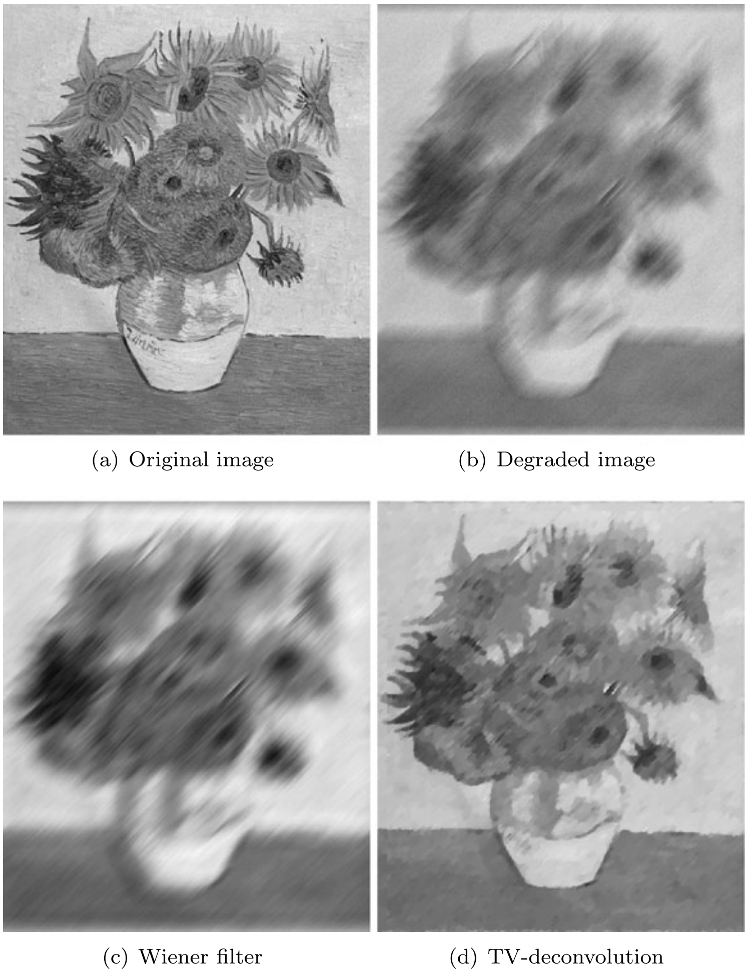
If the linear operator can not be implemented efficiently in this way, an alternative approach consists of additionally dualizing the functional with respect to $G(u)$, yielding

$$\begin{aligned} & \min_{u\in X}\max_{p\in Y, q\in X} -\langle u, \operatorname{div} p\rangle_X \\ & \quad + \langle Au - g, q\rangle_X - \delta_P(p) - \frac{1}{2\lambda}\|q\|^2, \end{aligned} \tag{74}$$

where $q \in X$ is the additional dual variable. In this case, we now have $F^*(p,q) = \delta_P(p) + \frac{1}{2\lambda}\|q\|^2$. Accordingly, the resolvent operator is given by

$$\begin{aligned} & (p,q) = (I + \sigma\partial F^*)^{-1}(\tilde{p},\tilde{q}) \\ \Longleftrightarrow \quad & p_{i,j} = \frac{\tilde{p}_{i,j}}{\max(1, |\tilde{p}_{i,j}|)}, \quad q_{i,j} = \frac{\tilde{q}_{i,j}}{1 + \sigma\lambda}. \end{aligned}$$

Figure 8 shows the application of the energy (72) to motion deblurring. While the classical Wiener filter is not able to restore the image, the total variation based approach yields a far better result. Figure 9 shows the application of (72) to zooming. One can observe that total variation based


(a) Original image (b) Degraded image

(c) Wiener filter (d) TV-deconvolution

Fig. 8 Motion deblurring using total variation regularization. (**a**) and (**b**) show the 400×470 clean image and a degraded version containing motion blur of approximately 30 pixels and Gaussian noise of standard deviation $\sigma = 0.01$. (**c**) is the result of standard Wiener filtering. (**d**) is the result of the total variation based deconvolution method. Note that the TV-based method yields visually much more appealing results

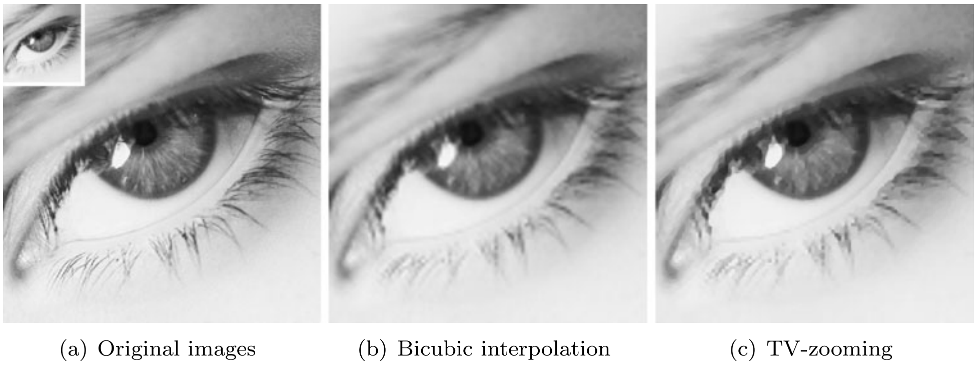

(a) Original images (b) Bicubic interpolation (c) TV-zooming

Fig. 9 Image zooming using total variation regularization. (**a**) shows the 384×384 original image and a by a factor of 4 downsampled version. (**b**) is the result of zooming by a factor of 4 using bicubic interpolation. (**c**) is the result of the total variation based zooming model. One can see that total variation based zooming yields much sharper image edges

zooming leads to a superresolved image with sharp boundaries whereas standard bicubic interpolation to a much more blurry result.

6.3.2 Image Inpainting

Image inpainting is the process of filling in lost image data. Although the total variation is useful for a number of applications, it is a rather weak prior for image inpainting. In the last years a lot of effort has been put into the development of more powerful image priors. An interesting class of priors is given by linear multi-level transformations such as wavelets, curvelets, etc. (see for example [14]). These transformations come along with the advantage of providing a compact representation of images while being computational efficient.

A generic model for image restoration can be derived from the classical ROF model (in the discrete setting), by simply replacing the gradient operator by a more general linear transform.

$$\min_{u\in X} \|\Phi u\|_1 + \frac{\lambda}{2}\|u-g\|_2^2, \tag{75}$$

where $\Phi : X \to W$ denotes the linear transform and $W = \mathbb{C}^K$ denotes the space of coefficients, usually some complex- or real-valued finite dimensional vector space. Here $K \in \mathbb{N}$, the dimension of W, may depend on different parameters such as the image size, the number of levels, orientations, etc.

Note that for unitary Φ (i.e. $\Phi^{-1} = \Phi^T$), the above model is equivalent to the following standard sparsity model:

$$\min_{x\in X} \|x\|_1 + \frac{\lambda}{2}\|\Phi^T x - g\|_2^2.$$

The L^1 norm $\|x\|_1$ is used to explicitly enforce sparsity in the coefficients x. We however found that model (75) leads to less artifacts in the solution and hence we propose to use this model.

The aim of model (75) is to find a sparse and hence compact representation of the image u in the domain of Φ, which has a small squared distance to the noisy data g. In particular, sparsity in the coefficients is attained by minimizing its L^1 norm. Clearly, minimizing the L^0 norm (i.e., the number of non-zero coefficients) would be better: but this problem is known to be NP-hard.

For the task of image inpainting, we consider a simple modification of (75). Let $D = \{(i,j),\ 1 \le i \le M,\ 1 \le j \le N\}$ denote the set of indices of the image domain and let $I \subset D$ denote the set of indices of the inpainting domain. The inpainting model can then be defined as

$$\min_{u\in X} \|\Phi u\|_1 + \frac{\lambda}{2}\sum_{i,j\in D\setminus I} (u_{i,j} - g_{i,j})^2. \tag{76}$$

Note that the choice $\lambda \in (0,\infty)$ corresponds to joint inpainting and denoising and the choice $\lambda = +\infty$ corresponds to pure inpainting.

The saddle-point formulation of (76) that fits into the general class of problems we are considering in this paper can be derived as

$$\min_{u\in X}\max_{c\in W} \langle \Phi u, c\rangle + \frac{\lambda}{2}\sum_{i,j\in D\setminus I} (u_{i,j} - g_{i,j})^2 - \delta_C(c), \tag{77}$$

where C is the convex set defined as

$$C = \{c \in W : \|c\|_\infty \le 1\}, \qquad \|c\|_\infty = \max_k |c_k|.$$

Let us identify in (77) $G(u) = \frac{\lambda}{2}\sum_{i,j\in D\setminus I}(u_{i,j} - g_{i,j})^2$ and $F^*(c) = \delta_C(c)$ in (77). Hence, the resolvent operators with respect to these functions can be easily evaluated.

$$c = (I + \sigma\partial F^*)^{-1}(\tilde{c}) \quad\Longleftrightarrow\quad c_k = \frac{\tilde{c}_k}{\max(1, |\tilde{c}_k|)},$$

and

$$u = (I + \tau\partial G)^{-1}(\tilde{u})$$
$$\Longleftrightarrow \quad u_{i,j} = \begin{cases} \tilde{u}_{i,j} & \text{if } (i,j) \in I \\ \frac{\tilde{u}_{i,j} + \tau\lambda g_{i,j}}{1+\tau\lambda} & \text{else.} \end{cases}$$

Since (77) is non-smooth we have to choose Algorithm 1 for optimization. Figure 10 shows the application of the inpainting model (76) to the recovery of lost lines (80% randomly chosen) of a color image. Figure 10 (c) shows the result when using $\Phi = \nabla$, i.e. the usual gradient operator. Figure 10 (d) shows the result but now using Φ to be the fast discrete curvelet transform [5]. One can see that the curvelet is much more successful in recovering long elongated structures and the smooth background structures. This example shows that different linear operators can be easily integrated in the proposed primal-dual algorithm.

6.3.3 *Motion Estimation*

Motion estimation (optical flow) is one of the central topics in imaging. The goal is to compute the *apparent* motion in image sequences. A typical variational formulation of total variation based motion estimation is given by (see e.g. [36])[5]

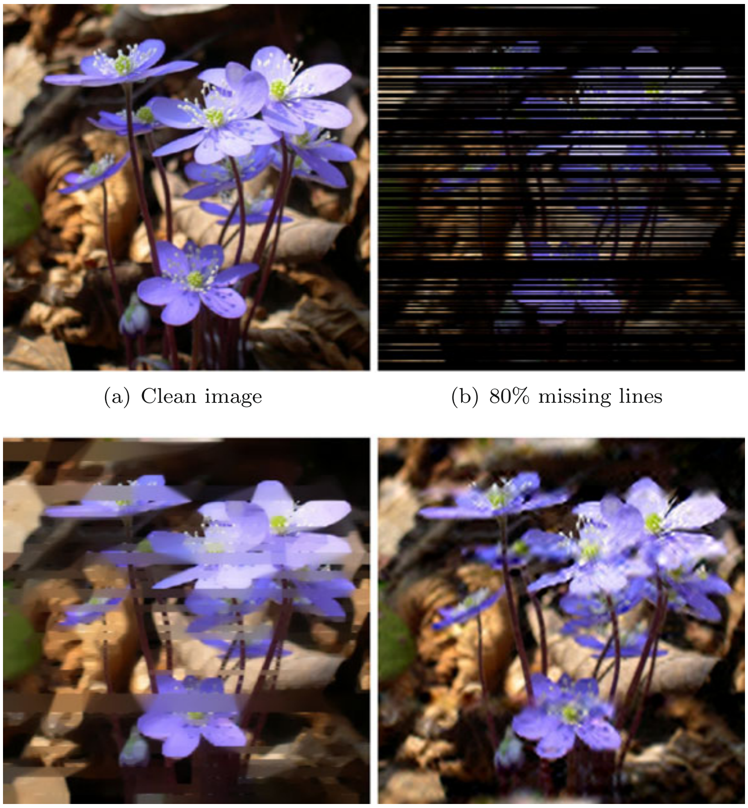
$$\min_v \int_\Omega |Dv| + \lambda\|\rho(v)\|_1, \tag{78}$$

where $v = (v_1, v_2)^T : \Omega \to \mathbb{R}^2$ is the motion field, and $\rho(v) = I_t + (\nabla I)^T(v - v^0)$ is the traditional optical flow constraint (OFC). It is obtained from a linearization of the assumption that the intensities of the pixels stay constant over time. I_t is the time derivative of the image sequence, ∇I is the spatial image gradient, and v^0 is some given motion field. The parameter λ is again used to defined the trade-off between data fitting and regularization.

In any practical situation, however, it is very unlikely (due to illumination changes and shadows), that the image intensities stay constant over time. This motivates the following slightly improved OFC, which explicitly models the varying illumination by means of an additive function u (see e.g. [34]).

$$\rho(u, v) = I_t + (\nabla I)^T(v - v^0) + \beta u.$$

[5] Interestingly, total variation regularization appears in [34] in the context of motion estimation several years before it was popularized by Rudin, Osher and Fatemi in [33] for image denoising.



(a) Clean image

(b) 80% missing lines

(c) TV inpainting

(d) Curvelet inpainting

Fig. 10 Recovery of lost image information. (**a**) shows the 384×384 clean image, (**b**) shows the destroyed image, where 80% of the lines are lost, (**c**) shows the result of the inpainting model (76) using a total variation prior and (**d**) shows the result when using a curvelet prior

The function $u : \Omega \to \mathbb{R}$ is expected to be smooth and hence we also regularize u by means of the total variation. The parameter β controls the influence of the illumination term. The improved motion estimation model is then given by

$$\min_{u,v} \int_\Omega |Du| + \int_\Omega |Dv| + \lambda \|\rho(u,v)\|_1.$$

Note that the OFC is valid only for small motion $(v - v^0)$. In order to account for large motion, the entire approach has to be integrated into a coarse-to-fine framework in order to re-estimate v^0. See again [36] for more details.

After discretization, we obtain the following primal motion model formulation

$$\min_{u \in X, v \in Y} \|\nabla u\|_1 + \|\nabla v\|_1 + \lambda \|\rho(u,v)\|_1, \tag{79}$$

where the discrete version of the OFC is now given by

$$\rho(u_{i,j}, v_{i,j}) = (I_t)_{i,j} + (\nabla I)^T_{i,j}(v_{i,j} - v^0_{i,j}) + \beta u_{i,j}.$$

The vectorial gradient $\nabla v = (\nabla v_1, \nabla v_2)$ is in the space $Z = Y \times Y$ equipped with a scalar product

$$\langle q, r \rangle_Z = \sum_{i,j} q^1_{i,j} r^1_{i,j} + q^2_{i,j} r^2_{i,j} + q^3_{i,j} r^3_{i,j} + q^4_{i,j} r^4_{i,j},$$
$$q = (q^1, q^2, q^3, q^4), \qquad r = (r^1, r^2, r^3, r^4) \in Z,$$

and a norm

$$\|\nabla v\|_1 = \sum_{i,j} |\nabla v_{i,j}|,$$
$$|\nabla v_{i,j}| = \sqrt{((\nabla v_1)^1_{i,j})^2 + ((\nabla v_1)^2_{i,j})^2 + ((\nabla v_2)^1_{i,j})^2 + ((\nabla v_2)^2_{i,j})^2}.$$

The saddle-point formulation of the primal motion estimation model (79) is obtained as

$$\min_{u \in X, v \in Y} \max_{p \in Y, q \in Z} \langle \nabla u, p \rangle_Y + \langle \nabla v, q \rangle_Z + \lambda \|\rho(u,v)\| - \delta_P(p) - \delta_Q(q), \tag{80}$$

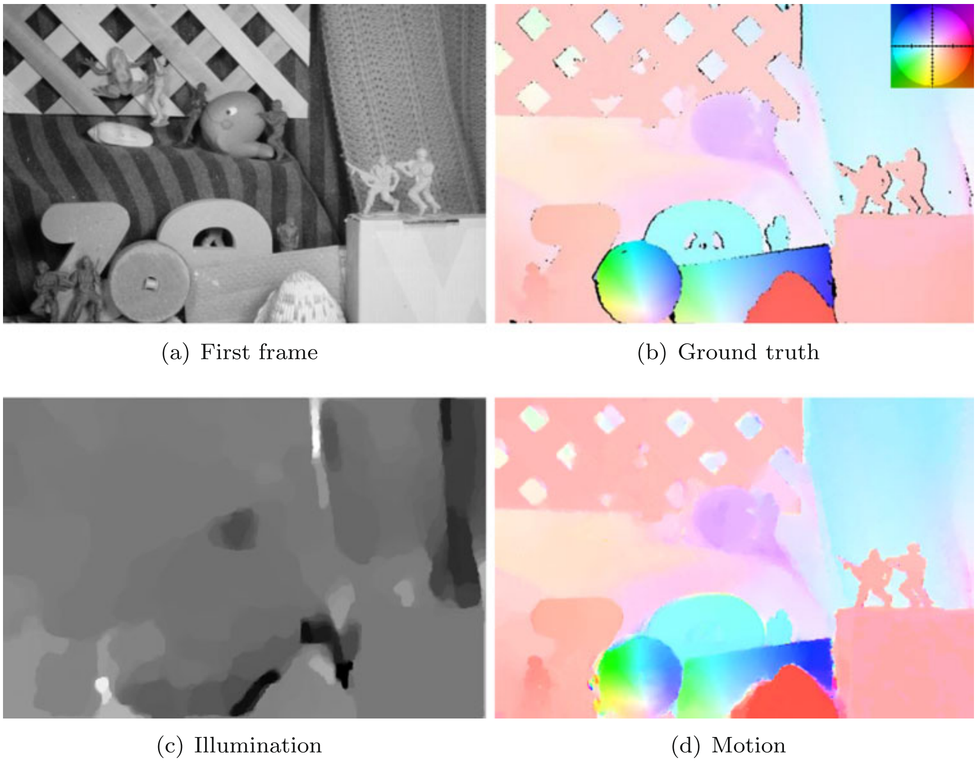
where the convex set P is defined as in (64) and the convex set Q is defined as

$$Q = \{ q \in Z : \|q\|_\infty \le 1 \},$$

and $\|q\|_\infty$ is the discrete maximum norm defined in Z as

$$\|q\|_\infty = \max_{i,j} |q_{i,j}|,$$
$$|q_{i,j}| = \sqrt{(q^1_{i,j})^2 + (q^2_{i,j})^2 + (q^3_{i,j})^2 + (q^4_{i,j})^2}.$$

Let us observe that (80) is a non-smooth convex problem with $G(u,v) = \lambda \|\rho(u,v)\|_1$ and $F^*(p,q) = \delta_P(p) +$



(a) First frame

(b) Ground truth

(c) Illumination

(d) Motion

Fig. 11 Motion estimation using total variation regularization and explicit illumination estimation. (**a**) shows one of two 584 × 388 input images and (**b**) shows the color coded ground truth motion field (black pixels indicate unknown motion vectors). (**c**) and (**d**) shows the estimated illumination and the color coded motion field

$\delta_Q(q)$. Hence we will have to rely on the basic Algorithm 1 to minimize (79).

Let us now describe the resolvent operators needed for the implementation of the algorithm. The resolvent operator with respect to $F^*(p,q)$ is again given by simple pointwise projections onto L^2 balls.

$$(p,q) = (I+\sigma\partial F^*)^{-1}(\tilde p,\tilde q)$$
$$\iff \quad p_{i,j} = \frac{\tilde p_{i,j}}{\max(1,|\tilde p_{i,j}|)}, \quad q_{i,j} = \frac{\tilde q_{i,j}}{\max(1,|\tilde q_{i,j}|)}.$$

Next, we give the resolvent operator with respect to $G(u,v)$. First, it is convenient to define $a_{i,j} = (\beta, (\nabla I)_{i,j})$ and $|a|^2_{i,j} = \beta^2 + |\nabla I|^2_{i,j}$. The solution of the resolvent operator is then given by

$$(u,v) = (I+\tau\partial G)^{-1}(\tilde u,\tilde v)$$
$$\iff \quad (u_{i,j},v_{i,j}) = (\tilde u_{i,j},\tilde v_{i,j})$$
$$+ \begin{cases} \tau\lambda a_{i,j} & \text{if } \rho(\tilde u_{i,j},\tilde v_{i,j}) < -\tau\lambda|a|^2_{i,j} \\ -\tau\lambda a_{i,j} & \text{if } \rho(\tilde u_{i,j},\tilde v_{i,j}) > \tau\lambda|a|^2_{i,j} \\ -\rho(\tilde u_{i,j},\tilde v_{i,j})a_{i,j}/|a|^2_{i,j} & \text{if } |\rho(\tilde u_{i,j},\tilde v_{i,j})| \le \tau\lambda|a|^2_{i,j}. \end{cases}$$

Figure 11 shows the results of applying the motion estimation model with explicit illumination estimation to the *Army* sequence from the Middlebury optical flow benchmark data set (http://vision.middlebury.edu/flow/). We integrated the algorithm into a standard coarse-to-fine framework in order to re-estimate v^0. The parameters of the model were set to $\lambda = 40$ and $\beta = 0.01$. One can see that the estimated motion field is very close to the ground truth motion field. Furthermore, one can see that illumination changes and shadows are well captured by the model (see for example the shadow on the left side of the shell). We have additionally implemented the algorithm on dedicated graphics hardware. This leads to a real-time performance of 30 frames per second for 640 × 480 images.

6.3.4 *Image Segmentation*

Finally, we consider the problem of finding a partition of an image into k pairwise disjoint regions, which minimizes the total interface between the sets, as for example in the piecewise constant Mumford-Shah problem [21]

$$\min_{(R_l)_{l=1}^k,(c_l)_{l=1}^k} \frac{1}{2}\sum_{l=1}^k Per(R_l;\Omega)$$
$$+ \frac{\lambda}{2}\sum_{l=1}^k \int_{R_l} |g(x)-c_l|^2\,dx \qquad (81)$$

where $g:\Omega\to\mathbb{R}$ is the input image, $c_l \in \mathbb{R}$ are the optimal mean values and the regions $(R_l)_{l=1}^k$ form a partition of Ω, that is, $R_l \cap R_m = \emptyset$ if $l \neq m$ and $\bigcup_{l=1}^k R_l = \Omega$. The parameter λ is again used to balance the data fitting term and the length term. Of course, given the partition $(R_l)_{l=1}^k$, the optimal constant $c_l = \int_{R_l} g\,ds/|R_l|$ is the average value of g

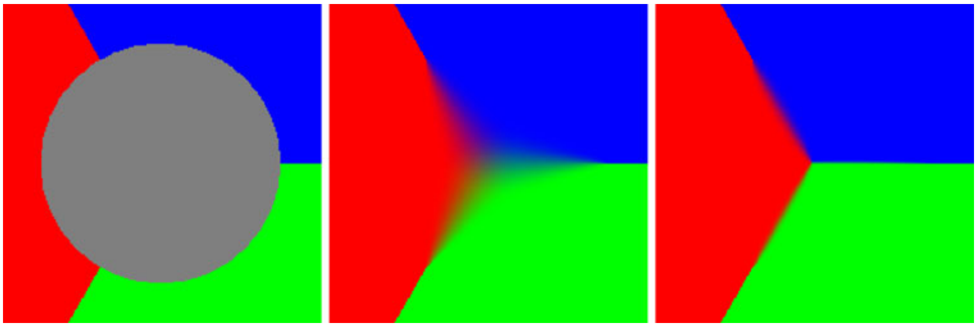

(a) Input image (b) Relaxation (83) (c) Relaxation (84)

Fig. 12 Triple junction experiment with $k=3$. (**a**) shows the 200×200 input image with given boundary datum. (**b**) shows the result using the relaxation (83), and (**c**) shows the result using the relaxation (84)

on R_l for each $l=1,\dots,k$. On the other hand, finding the minimum of (81) with respect to the partition $(R_l)_{l=1}^k$ is a hard task, even for fixed values $(c_l)_{l=1}^k$. It is known that its discrete counterpart (the *Pott's model*) is NP-hard, so that it is unlikely that (81) has a simple convex representation, at least without increasing drastically the number of variables. In the following, we will assume that the optimal mean values c_l are known and fixed, which leads us to consider the following generic representation of (81).

$$\begin{aligned}
&\min_{u=(u_l)_{l=1}^k} J(u) + \sum_{l=1}^k \int_\Omega u_l f_l \, dx, \\
&u_l(x) \ge 0, \qquad \sum_{l=1}^k u_l(x) = 1, \quad \forall x \in \Omega,
\end{aligned} \tag{82}$$

where $u=(u_l)_{l=1}^k : \Omega \to \mathbb{R}^k$ is the labeling function and $f_l = \lambda |g(x) - c_l|^2/2$ as in (81) or any other weighting function obtained from more sophisticated data terms (e.g., based on histograms, or a similarity measure in case of stereo). Different choices have been proposed for relaxations of the length term $J(u)$. The most straightforward relaxation as used in [35] is

$$J_1(u) = \frac{1}{2} \sum_{l=1}^k \int_\Omega |Du_l|, \tag{83}$$

which is simply the sum of the total variation of each labeling function u_i However, it can be shown that this relaxation is too small [8]. A better choice is (see Fig. 12)

$$\begin{aligned}
&J_2(u) = \int_\Omega \Psi(Du), \\
&\Psi(p) = \sup_q \{\langle p, q\rangle : |q_l - q_m| \le 1, \ 1 \le l < m \le k\},
\end{aligned} \tag{84}$$

where $p=(p_1,\dots,p_k)$ and $q=(q_1,\dots,q_k)$. This energy is also a sort of total variation but now defined on the complete vector-valued function u. This construction is related to the theory of *paired calibrations* [4, 18].

Let us now turn to the discrete setting. The primal-dual formulation of the partitioning problem (82) is obtained as

$$\begin{aligned}
&\min_{u=(u_l)_{l=1}^k} \max_{p=(p)_{l=1}^k} \left(\sum_{l=1}^k \langle \nabla u_l, p_l\rangle + \langle u_l, f_l\rangle \right) \\
&\quad + \delta_U(u) - \delta_P(p),
\end{aligned} \tag{85}$$

where $f=(f_l)_{l=1}^k \in X^k$ is the discretized weighting function, $u=(u_l)_{l=1}^k \in X^k$ is the primal variable, representing the assignment of each pixel to the labels and $p=(p_l)_{l=1}^k \in Y^k$ is the dual variable, which will be constrained to stay in a set P which we will soon make precise.

In the above formula, we can identify $G(u)=\delta_U(u)$, which forces the solution to stay in the unit simplex U, defined as

$$U = \left\{ u \in X^k : (u_l)_{i,j} \ge 0, \ \sum_{l=1}^k (u_l)_{i,j} = 1 \right\}.$$

Furthermore we can identify $F^*(p) = \delta_P(p)$, where the convex sets $P = P_1$ or P_2 either realizes the standard relaxation $J_1(u)$ or the stronger relaxation $J_2(u)$ of the total interface surface. In particular, the set P_1 arises from an application of the dual formulation of the total variation to each vector u_l,

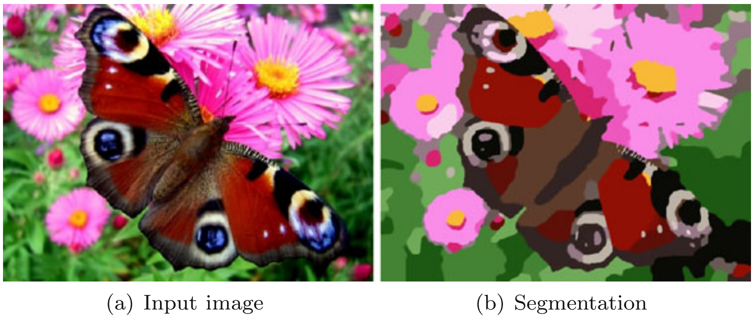
$$P_1 = \left\{ p \in Y^k : \|p_l\|_\infty \le \frac{1}{2} \right\}.$$

On the other hand, the set P_2 is directly obtained from the definition of relaxation (84),

$$P_2 = \left\{ p \in Y^k : \|p_l - p_m\|_\infty \le 1, \ 1 \le l < m \le k \right\},$$

which is essentially an intersection of unit balls.

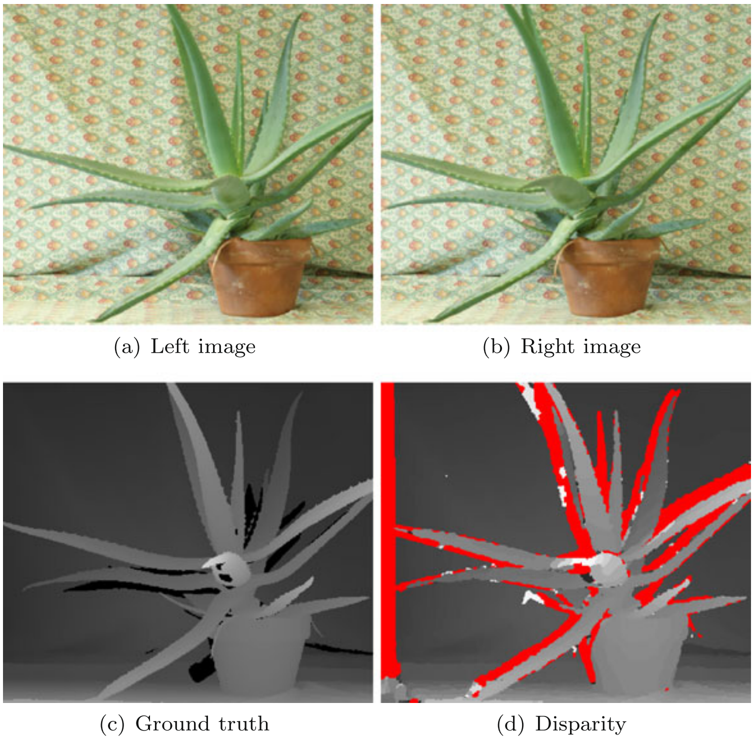
Next, we detail the resolvent operators. The resolvent operator with respect to G is an orthogonal projector onto the unit simplex defined by the convex set U. It is known that this projection can be performed in a finite number of steps. See for example [20] for an algorithm based on successive projections and corrections.



(a) Input image

(b) Segmentation

Fig. 13 Piecewise constant Mumford-Shah segmentation of a natural image with $k = 16$. (**a**) shows the 580×435 input image and (**b**) is the minimizer of energy (82)



(a) Left image

(b) Right image

(c) Ground truth

(d) Disparity

Fig. 14 Stereo estimation using the Potts model. The *first row* shows the *left* and *right* input images of size 427×370, downloaded from (http://vision.middlebury.edu/stereo/). The *second row* show the ground truth disparity image (*black pixels* indicate unknown disparity values) and the estimated disparity values of the segmentation model (*red pixels* indicate occluded pixels)

The resolvent operator with respect to F^* is also an orthogonal projector. In case of P_1 the projection is very easy, since it reduces to pointwise projections onto unit balls. In case of P_2 the projection is more complicated, since the complete vector $p_{i,j}$ is projected onto an intersection of convex sets. This can for example be performed by Dykstra's algorithm [3]. Finally we adhere that since (85) is non-smooth, we have to use Algorithm 1 to minimize the segmentation model.

Figure 12 shows the result of different relaxations for the triple-junction experiment. Here, the task is to complete the segmentation boundary in the gray area, where the weighting function g_l is set to zero. One can see that the simple relaxation (83) leads to a non-integer (binary, with $u \in \{0, 1\}^k$ a.e.) solution. On the other hand, the stronger relaxation (84) yields an almost binary solution and hence a globally optimal solution of the segmentation model.

Figure 13 shows the result of piecewise constant Mumford-Shah segmentation (81) using the relaxation $J_2(u)$. We used $k = 16$ labels and the mean color values c_i have been initialized using k-means clustering. The regularization parameter was set to $\lambda = 5$. Note that again, the result is almost binary.

Finally, Fig. 14 shows the application of the Potts model to disparity estimation. Here, the labels correspond to distinct disparity (and hence depth) values. The weighting functions f_l are computed based on a pixel-wise similarity measure between the stereo images for a given disparity value associated with the label l. We use the sum of the absolute differences between the image gradients of the left and the right images to achieve some invariance to intensity changes. Furthermore, we introduce an additional label endowed with a small constant weighting function to account for occluded pixels.

In all, we obtain a Potts model with approximately 200 labels, which in our example amounts to solve an optimization problem of more than 30 million unknowns. Due to memory restrictions, we apply the weaker relaxation (83), which however gives satisfactory results for this example. Our GPU-based implementation of the primal-dual algorithm finds an approximate solution (with reasonable accuracy) in less than 30 seconds. This shows that the algorithm can be applied to very large convex optimization problems.

7 Discussion

In this paper we have proposed a first-order primal-dual algorithm and showed how it can be useful for solving efficiently a large family of convex problems arising in imaging. We have shown that it can be adapted to yield an accelerated rate of convergence depending of the regularity of the problem, optimal with respect to what is know in the scientific literature. In particular, the algorithm converges with $O(1/N)$ for non-smooth problems, with $O(1/N^2)$ for problems where either the primal or dual objective is uniformly convex, and that it converges linearly, i.e. like $O(\omega^N)$ for $\omega < 1$, for "smooth" problems (where both the primal and dual are uniformly convex). Our theoretical results are supported by the numerical experiments.

There are still several interesting questions, which need to be addressed in the future: (a) the case where the linear operator K is unbounded or has a large (in general unknown) norm, as it is usually the case in infinite dimension or in finite elements discretization of continuous problems; (b) how to automatically determine the smoothness parameters or to locally adapt to the regularity of the objective; (c) understand why does the Arrow-Hurwicz method perform so well in some situations.

Acknowledgements Antonin Chambolle is partially supported by the Agence Nationale de la Recherche (grant "MICA" ANR-08-BLAN-0082). Thomas Pock is partially supported by the Austrian Science Fund (FWF) under the doctoral program "Confluence of Vision and Graphics" W1209. The authors wish to thank Daniel Cremers and Marc Teboulle for helpful discussion. Also, we are very grateful to Marc Teboulle and the anonymous referees who carefully read a first version of this work and provided us we very helpful comments.

References

1. Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in linear and nonlinear programming. In: Chenery, H.B., Johnson, S.M., Karlin, S., Marschak, T., Solow, R.M. (eds.) Stanford Mathematical Studies in the Social Sciences, vol. II. Stanford University Press, Stanford (1958)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
3. Boyle, J.P., Dykstra, R.L.: A method for finding projections onto the intersection of convex sets in Hilbert spaces. In: Advances in Order Restricted Statistical Inference, Iowa City, Iowa, 1985. Lecture Notes in Statist., vol. 37, pp. 28–47. Springer, Berlin (1986)
4. Brakke, K.A.: Soap films and covering spaces. J. Geom. Anal. **5**(4), 445–514 (1995)
5. Candès, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. Multiscale Model. Simul. **5**(3), 861–899 (2006) (electronic)
6. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1–2), 89–97 (2004). Special issue on mathematics and image analysis
7. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 136–152 (2005)
8. Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Technical Report 649, CMAP, Ecole Polytechnique, France (2008)
9. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. http://hal.archives-ouvertes.fr/hal-00490826 (2010)
10. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program., Ser. A **55**(3), 293–318 (1992)
11. Esser, E.: Applications of Lagrangian-based alternating direction methods and connections to split Bregman. CAM Reports 09-31, UCLA, Center for Applied Math. (2009)
12. Esser, E., Zhang, X., Chan, T.: A general framework for a class of first order primal-dual algorithms for tv minimization. CAM Reports 09-67, UCLA, Center for Applied Math. (2009)
13. Fadili, J., Peyré, G.: Total variation projection with first order schemes. http://hal.archives-ouvertes.fr/hal-00380491/ (2009)
14. Fadili, J., Starck, J.-L., Elad, M., Donoho, D.: Mcalab: Reproducible research in signal and image decomposition and inpainting. Comput. Sci. Eng. **12**(1), 44–63 (2010)
15. Goldstein, T., Osher, S.: The split Bregman algorithm for l1 regularized problems. CAM Reports 08-29, UCLA, Center for Applied Math. (2008)
16. He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for total variation image restoration. Technical Report 2790, Optimization Online, November 2010 (available at www.optimization-online.org)
17. Korpelevič, G.M.: An extragradient method for finding saddle points and for other problems. Èkon. Mat. Metody **12**(4), 747–756 (1976)
18. Lawlor, G., Morgan, F.: Paired calibrations applied to soap films, immiscible fluids, and surfaces or networks minimizing other norms. Pac. J. Math. **166**(1), 55–83 (1994)
19. Lions, P.-L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**(6), 964–979 (1979)
20. Michelot, C.: A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbf{R}^n$. J. Optim. Theory Appl. **50**(1), 195–200 (1986)
21. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**, 577–685 (1989)
22. Nedić, A., Ozdaglar, A.: Subgradient methods for saddle-point problems. J. Optim. Theory Appl. **142**(1), 1 (2009)
23. Nemirovski, A.: Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM J. Optim. **15**(1), 229–251 (2004) (electronic)
24. Nemirovski, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. A Wiley-Interscience Publication. Wiley, New York (1983). Translated from the Russian and with a preface

by E.R. Dawson, Wiley-Interscience Series in Discrete Mathematics

25. Nesterov, Yu.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Dokl. Akad. Nauk SSSR **269**(3), 543–547 (1983)
26. Nesterov, Yu.: Introductory Lectures on Convex Optimization. Applied Optimization, vol. 87. Kluwer Academic, Boston (2004). A basic course
27. Nesterov, Yu.: Smooth minimization of non-smooth functions. Math. Program., Ser. A **103**(1), 127–152 (2005)
28. Nesterov, Yu.: Gradient methods for minimizing composite objective function. Technical report, CORE DISCUSSION PAPER (2007)
29. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford-Shah functional. In: ICCV Proceedings, LNCS. Springer, Berlin (2009)
30. Popov, L.D.: A modification of the Arrow-Hurwitz method of search for saddle points. Mat. Zametki **28**(5), 777–784, 803 (1980)
31. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. **14**(5), 877–898 (1976)
32. Rockafellar, R.T.: Convex Analysis, Princeton Landmarks in Mathematics. Princeton University Press, Princeton (1997). Reprint of the 1970 original, Princeton Paperbacks
33. Rudin, L., Osher, S.J., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992) [also in Experimental Mathematics: Computational Issues in Nonlinear Science (Proc. Los Alamos Conf. 1991)]
34. Shulman, D., Hervé, J.-Y.: Regularization of discontinuous flow fields. In: Proceedings Workshop on Visual Motion, pp. 81–86 (1989)
35. Zach, C., Gallup, D., Frahm, J.M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: Vision, Modeling, and Visualization 2008, pp. 243–252. IOS Press, Amsterdam (2008)
36. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L^1 optical flow. In: 29th DAGM Symposium on Pattern Recognition, pp. 214–223. Heidelberg, Germany (2007)
37. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. CAM Reports 08-34, UCLA, Center for Applied Math. (2008)