Noname manuscript No.
(will be inserted by the editor)

Automated Parameter Selection for Total Variation Minimization in Image Restoration

Andreas Langer

Received: date / Accepted: date

Abstract Algorithms for automatically selecting a scalar or locally varying regularization parameter for total variation models with an L^τ-data fidelity term, $\tau \in \{1, 2\}$, are presented. The automated selection of the regularization parameter is based on the discrepancy principle, whereby in each iteration a total variation model has to be minimized. In the case of a locally varying parameter this amounts to solve a multi-scale total variation minimization problem. For solving the constituted multi-scale total variation model convergent first and second order methods are introduced and analyzed. Numerical experiments for image denoising and image deblurring show the efficiency, the competitiveness, and the performance of the proposed fully automated scalar and locally varying parameter selection algorithms.

Keywords Total variation minimization · Locally dependent regularization parameter · Automated parameter selection · L^2-data fidelity · L^1-data fidelity · Discrepancy principle · Constrained/unconstrained problem · Gaussian noise · Impulse noise

1 Introduction

Observed images are often contaminated by noise and may be additionally distorted by some measurement device. Then the obtained data g can be described as

$$g = \mathcal{N}(T\hat{u}),$$

where $\hat{u}$ is the unknown original image, T is a linear bounded operator modeling the image-formation device, and $\mathcal{N}$ represents noise. In this paper, we consider images which are contaminated either by white Gaussian noise or impulse noise. While for white Gaussian noise the degraded image g is obtained as

$$g = T\hat{u} + \eta,$$

where the noise η is oscillatory with zero mean and standard deviation σ, there are two main models for impulse noise, that are widely used in a variety of applications, namely salt-and-pepper noise and random-valued impulse noise. We assume that $T\hat{u}$ is in the dynamic range $[0, 1]$, i.e., $0 \leq T\hat{u} \leq 1$, then in the presence of salt-and-pepper noise the observation g is given by

$$g(x) = \begin{cases} 0 & \text{with probability } r_1 \in [0, 1), \\ 1 & \text{with probability } r_2 \in [0, 1), \\ T\hat{u}(x) & \text{with probability } 1 - r_1 - r_2, \end{cases} \tag{1.1}$$

with $1 - r_1 - r_2 > 0$. If the image is contaminated by random-valued impulse noise, then g is described as

$$g(x) = \begin{cases} \rho & \text{with probability } r \in [0, 1), \\ T\hat{u}(x) & \text{with probability } 1 - r, \end{cases} \tag{1.2}$$

where ρ is a uniformly distributed random variable in the image intensity range $[0, 1]$.

The recovery of $\hat{u}$ from the given degraded image g is an ill-posed inverse problem and thus regularization techniques are required to restore the unknown image [36]. A good approximation of $\hat{u}$ may be obtained by solving a minimization problem of the type

$$\min_u \mathcal{H}(u; g) + \alpha\mathcal{R}(u), \tag{1.3}$$

where $\mathcal{H}(.; g)$ represents a data fidelity term, which enforces the consistency between the recovered and measured image, $\mathcal{R}$ is an appropriate filter or regularization

A. Langer
Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
E-mail: andreas.langer@mathematik.uni-stuttgart.de

term, which prevents over-fitting, and $\alpha > 0$ is a regularization parameter weighting the importance of the two terms. We aim at reconstructions in which edges and discontinuities are preserved. For this purpose we use the total variation as a regularization term, first proposed in [68] for image denoising. We recall, that for $u \in L^1(\Omega)$, where $\Omega \subset \mathbb{R}^2$ is a bounded open domain with Lipschitz boundary,

$$\int_\Omega |Du| = \sup \left\{ \int_\Omega u \operatorname{div} \boldsymbol{\xi} \, dx : \boldsymbol{\xi} \in [C_c^1(\Omega)]^2, \, \|\boldsymbol{\xi}\|_{L^\infty(\Omega)} \leq 1 \right\}$$

is the variation of u in Ω. Here, $L^q(\Omega)$, with $q \in [1, \infty]$, denotes the usual Lebesgue space [1] and $C_c^l(\Omega)$, $l \in \mathbb{N}$, is the space of l-times continuously differentiable functions with compact support in Ω. Further, $BV(\Omega)$ denotes the space of functions with bounded variation, i.e., $u \in BV(\Omega)$ if and only if $u \in L^1(\Omega)$ and $\int_\Omega |Du| < \infty$. In this case, we call $\int_\Omega |Du|$ the total variation of u in Ω; see [4,40] for more details. If $u \in W^{1,1}(\Omega)$, then $\int_\Omega |Du| = \int_\Omega |\nabla u| dx$. Here and in the remaining of the paper we choose $\mathcal{R}(u) = \int_\Omega |Du|$. However, we note that other regularization terms, as the total generalized variation [13], the non-local total variation [52], the Mumford-Shah regularizer [59], or higher order regularizers (see e.g. [65] and references therein) might be used as well.

The choice of $\mathcal{H}$ typically depends on the type of noise contamination. For images corrupted by Gaussian noise a quadratic L^2-data fidelity term is typically chosen and has been successfully used; see for example [16,17,18,21,24,25,26,28,29,41,60,64,79,81]. In this approach, which we refer to as the L^2-TV model, the image $\hat{u}$ is recovered from the observed data g by solving

$$\min_{u \in BV(\Omega)} \frac{1}{2} \|Tu - g\|_{L^2(\Omega)}^2 + \alpha \mathcal{R}(u). \tag{1.4}$$

In the presence of impulse noise, e.g., salt-and-pepper noise or random-valued impulse noise, the above model usually does not yield a satisfactory restoration. In this context, a more successful approach, suggested in [2,62, 63], uses a non-smooth L^1-data fidelity term instead of the L^2-data fidelity term in (1.4), i.e., one considers

$$\min_{u \in BV(\Omega)} \|Tu - g\|_{L^1(\Omega)} + \alpha \mathcal{R}(u), \tag{1.5}$$

which we call the L^1-TV model. In this paper, we are interested in both models, i.e., the L^2-TV and the L^1-TV model, and condense them into

$$\min_{u \in BV(\Omega)} \mathcal{H}_\tau(u; g) + \alpha \mathcal{R}(u) \tag{1.6}$$

to obtain a combined model for removing Gaussian or impulsive noise, where $\mathcal{H}_\tau(u; g) := \frac{1}{\tau} \|Tu - g\|_{L^\tau(\Omega)}^\tau$ for $\tau = 1, 2$. Note that instead of (1.6) one can consider the equivalent problem

$$\min_{u \in BV(\Omega)} \lambda \mathcal{H}_\tau(u; g) + \mathcal{R}(u), \tag{1.7}$$

where $\lambda = \frac{1}{\alpha} > 0$. For images which are simultaneously contaminated by Gaussian and impulse noise a combined L^1-L^2-data fidelity term has been recently suggested and demonstrated to work satisfactory [48]. However, in this paper, we concentrate on images degraded by only one type of noise, i.e., either Gaussian noise or one type of impulse noise, and perhaps additionally corrupted by some measurement device.

For the reconstruction of such images the proper choice of α in (1.6) and λ in (1.7) is delicate. In particular, large α and small λ, which lead to an over-smoothed reconstruction, not only remove noise but also eliminate details in images. On the other hand, small α and large λ lead to solutions which fit the given data properly but therefore retain noise in homogeneous regions. Hence a good reconstruction can be obtained by choosing α and respectively λ such that a good compromise of the aforementioned effects are made. There are several ways of how to select α in (1.6) and equivalently λ in (1.7), such as manually by the trial-and-error method, the unbiased predictive risk estimator method (UPRE) [57], the Stein unbiased risk estimator method (SURE) [69, 32,10], the generalized cross-validation method [42,55, 56], the L-curve method [44,45], the discrepancy principle [58], and the variational Bayes' approach [6]. Further parameter selection methods for general inverse problems can be found for example in [36,37,75,76]

The UPRE is based on a statistical estimator of the mean squared norm of the predicitve error and only valid if the regularized solution depends linearly on the data [77]. An extension to the L^2-TV model is considered in [56].

The SURE provides an unbiased estimate of the mean-squared error between the noiseless image and the regularized solution. While the original method is restricted to the case of denoising images contaminated by additive white Gaussian noise, its generalization, named GSURE (generalized SURE), is developed to deal with a more diverse set of inverse problems [34, 39].

The generalized cross-validation evaluation formula can be computed when the regularization term is of quadratic form. However, it cannot be derived explicitly for (1.6) and (1.7) due to the nonlinearity of the total variation. The utilization of a variable-splitting method and an alternating minimization scheme in [55]

allows to use the generalized cross-validation technique to estimate the regularization parameter for the L^2-TV model and the L^1-TV model.

The L-curve method is based on a parameteric plot of the norm of the regularized solution versus the norm of the corresponding residual. This plot seems to produce an L-shaped curve and the regularization parameter is chosen to be the corner point of this curve. However, in order to obtain the L-curve many times for different parameters (1.6) or (1.7) has to be solved, which makes this method computationally expensive. Moreover, it has been demonstrated that the L-curve alone should not be the basis for a convergent parameter choice strategy, since it fails to converge to the original image when the noise-level tends to zero; see [35,43,76].

The variational Bayes' approach in [6] relies on a stochastic method approximating a posteriori distribution by a product of distributions utilizing Kullback-Leibler divergence.

Applying the discrepancy principle to estimate α in (1.6) or λ in (1.7), the image restoration problem can be formulated as a constrained optimization problem of the form

$$\min_{u\in BV(\Omega)} \mathcal{R}(u) \quad \text{subject to (s.t.)} \quad \mathcal{H}_\tau(u;g) = \mathcal{B}_\tau \tag{1.8}$$

where $\mathcal{B}_\tau := \frac{\nu_\tau}{\tau}|\Omega|$ with $\nu_\tau > 0$ being here a constant depending on the underlying noise, $\tau = 1, 2$, and $|\Omega|$ denoting the volume of Ω; see Section 4 for more details. Note, that the constrained minimization problem (1.8) is naturally linked to the unconstrained minimization problem (1.7) and accordingly to (1.6). In particular, there exists a constant $\lambda \geq 0$ such that the unconstrained problem (1.7) is equivalent to the constrained problem (1.8) if T does not annihilate constant functions, i.e., $T \in \mathcal{L}(L^2(\Omega))$ is such that $T \cdot 1 = 1$; see Section 2 for more details. Several methods based on the discrepancy principle and problem (1.8) with $\tau = 2$ have been proposed in the literature, see for example [9, 16,39,46,80] and references therein, while not so much attention has been given to the case $\tau = 1$, see for example [61,79]. Our first contribution of this paper is to present a method which automatically computes the regularization parameter α in (1.6) based on (1.8) for $\tau = 1$ as well as for $\tau = 2$. Our approach is motivated by the parameter selection algorithm presented in [16] which was originally introduced for L^2-TV image denoising only, i.e., when $T = I$ where I denotes the identity operator. In this setting the algorithm in [16] is shown to converge to a parameter α^* such that the corresponding minimizer u_{α^*} of (1.6) is also a solution of (1.8). The proof relies on the non-increase of the function $\alpha \mapsto \frac{\mathcal{H}_2(u_\alpha;g)}{\mathcal{B}_2}$. However, this important property does not hold for operators $T \neq I$ in general. Nevertheless, we generalize the algorithm from [16] to problems of the type (1.6) for $\tau = 1, 2$ and for general linear bounded operators T, e.g., T might be a convolution type of operator. Here as in [16] the automated adjustment of α relies on the constraint in (1.8), where the main idea is, that whenever $\frac{\mathcal{B}_\tau}{\mathcal{H}_\tau(u_\alpha;g)} < 1$, then α is decreased to decrease $\mathcal{H}_\tau(u_\alpha;g)$. Similarly, α is increased whenever $\frac{\mathcal{B}_\tau}{\mathcal{H}_\tau(u_\alpha;g)} > 1$ to increase $\mathcal{H}_\tau(u_\alpha;g)$. Using this scheme and utilizing an appropriate update of α, which is different than the one used in [16], we are able to show analytically and numerically that our approach indeed converges to the desired regularization parameter. Further, besides the general applicability of our proposed method it even possesses advantages for the case $\tau = 2$ and $T = I$ over the algorithm from [16] with respect to convergence. More precisely, in our numerics it turned out that our proposed method always needs less or at least the same number of iterations as the algorithm from [16] till termination.

Note, that a scalar regularization parameter might not be the best choice for every image restoration problem, since images usually have large homogeneous regions as well as parts with a lot of details. Actually it seems obvious that α should be small, or λ should be large, in parts with small features in order to preserve the details. On the contrary α should be large, or λ should be small, in homogeneous parts to remove noise considerable. With such a choice of a spatially varying weight we expect better reconstructions than with a globally constant parameter, as demonstrated for example in [31,50]. This motivated to consider multi-scale total variation models with spatially varying parameters initially suggested in [67]. The multi-scale version of (1.6) reads as

$$\min_{u\in BV(\Omega)} \mathcal{H}_\tau(u;g) + \int_\Omega \alpha(x)|Du| \tag{1.9}$$

while for (1.7) one writes

$$\min_{u\in BV(\Omega)} \frac{1}{\tau}\int_\Omega \lambda(x)|Tu-g|^\tau dx + \mathcal{R}(u), \tag{1.10}$$

and in the sequel we refer to (1.9) and (1.10) as the multi-scale L^τ-TV model.

In [71] the influence of the scale of an image feature on the choice of α is studied and the obtained observations were later used in [70] to construct an updating scheme of α. Based on (1.10) in [8] a piecewise constant function λ, where the pieces are defined by a partitioning of the image due to a pre-segmentation, is determined. In particular, for each segment a scalar

λ_i, $i = 1, \dots, \#\text{pieces}$ is computed by Uzawa's method [23].

Later it was noticed that stable choices of λ respectively α should incorporate statistical properties of the noise. In this vein, in [3,31,38] for the problem (1.10) automated update rules for λ based on statistics of local constraints were proposed. In [38] a two level approach for variational denoising is considered, where in the first level noise and relevant texture are isolated in order to compute local constraints based on local variance estimation. In the second level a gradient descent method and an update formula for $\lambda(x)$ derived from the Euler-Lagrange equation is utilized. An adaptation of this approach to multiplicative noise can be found in [54]. For convolution type of problems in [3] based on an estimate of the noise variance for each pixel an automatic updating scheme of λ using Uzawa's method is created. This approach is improved in [31] by determining the fidelity weights due to the Gumbel statistic for the maximum of a finite number of random variables associated with localized image residuals and by incorporating hierarchical image decompositions, proposed in [73,74], to speed up the iterative parameter adjustment process. An adaptation of this approach to a total variation model with L^1 local constraints is studied in [50]. A different approach has been proposed in [72] for image denoising only, where non-local means [14] are used to create a non-local data fidelity term. While in all these approaches the adjustment of λ relies on the output of T being a deteriorated image again, in [49] the method of [31] is adjusted to the situation where T is an orthogonal wavelet transform or Fourier transform.

Motivated by multi-scale total variation minimization, the second contribution of this paper is concerned with the automated selection of a suitable spatially varying α for the optimization problem in (1.9) for $\tau = 1, 2$. Based on our considerations for an automatic scalar regularization parameter selection, we present algorithms where the adjustment of a locally varying α is fully automatic. Differently to the scalar case the adjustment of α is now based on local constraints, similarly as already considered for example in [3,31,50]. However, our approach differs significantly from these previous works, where problem (1.10) is considered and Uzawa's method or an Uzawa-like method is utilized for the update of the spatially varying parameter. Note, that in Uzawa's method an additional parameter has to be introduced and chosen accordingly. We propose an update-scheme of α which does not need any additional parameter and hence is not similar to Uzawa's method. Moreover, differently to the approaches in [31, 50] where the initial regularization parameter $\lambda > 0$ has to be set sufficiently small, in our approach any initial $\alpha > 0$ is allowed. In this sense is our algorithm even more general than the ones presented in [31,50].

The outline of the remaining of the paper is as follows. In Section 2 we revisit and discuss the connection between the constrained minimization problem (1.8) and the unconstrained optimization problem (1.6). The basic notations and definitions for a discrete setting are given in Section 3. Section 4 is devoted to the automated scalar parameter selection. In particular, we present our proposed method and analyze its convergence behavior. Based on local constraints we describe in Section 5 our new locally adapted total variation algorithm in detail. Algorithms for performing total variation minimization for spatially varying α are presented in Section 6 where also their convergence properties are studied. To demonstrate the performance of the new algorithms we present in Section 7 numerical experiments for image denoising and image deblurring. Finally, in Section 8 conclusions are drawn.

2 Constrained versus unconstrained minimization problem

In this section we explain the connection between the unconstrained minimization problem (1.6) and the constrained optimization problem (1.8). For this purpose we introduce the following basic terminology. Let $\mathcal{V}$ be a locally convex space, $\mathcal{V}'$ its topological dual, and $\langle \cdot, \cdot \rangle$ the bilinear canonical pairing over $\mathcal{V} \times \mathcal{V}'$. The domain of a functional $\mathcal{J} : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is defined as the set

$$\mathrm{Dom}(\mathcal{J}) := \{v \in \mathcal{V} \ : \ \mathcal{J}(v) < \infty\}.$$

A functional $\mathcal{J}$ is called *lower semicontinuous* (l.s.c) if for every weakly convergent subsequence $v^{(n)} \rightharpoonup \hat{v}$ we have

$$\liminf_{v^{(n)} \rightharpoonup \hat{v}} \mathcal{J}(v^{(n)}) \geq \mathcal{J}(\hat{v}).$$

For a convex functional $\mathcal{J} : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, we define the *subdifferential* of $\mathcal{J}$ at $v \in \mathcal{V}$, as the set valued function $\partial\mathcal{J}(v) = \emptyset$ if $\mathcal{J}(v) = \infty$, and otherwise as

$$\partial\mathcal{J}(v) = \{v^* \in \mathcal{V}' \ : \ \langle v^*, u - v\rangle + \mathcal{J}(v) \leq \mathcal{J}(u) \ \forall u \in \mathcal{V}\}.$$

Moreover, for any operator T we denote by T^* its adjoint and by $\mathcal{L}(L^2(\Omega))$ we denote the space of linear and continuous operators from $L^2(\Omega)$ to $L^2(\Omega)$.

Theorem 2.1 *Assume that $T \in \mathcal{L}(L^2(\Omega))$ does not annihilate constant functions, i.e., $T1_\Omega \neq 0$, where $1_\Omega(x) = 1$ for $x \in \Omega$. Then the problem*

$$\min_{u \in BV(\Omega)} \mathcal{R}(u) \quad s.t. \quad \mathcal{H}_\tau(u; g) \leq \mathcal{B}_\tau \tag{2.1}$$

has a solution for $\tau = 1, 2$.

Proof For a proof we refer the reader to [18] and [50]. □

Moreover, we have the following statement.

Proposition 2.1 *Assume that $T \in \mathcal{L}(L^2(\Omega))$ is such that $T \cdot 1 = 1$ and $\nu_\tau|\Omega| \leq \|g - \frac{1}{|\Omega|}\int_\Omega g\|^\tau_{L^\tau(\Omega)}$. Then problem* (2.1) *is equivalent to the constrained minimization problem* (1.8) *for $\tau = 1, 2$.*

Proof For $\tau = 2$ the statement is shown in [18]. We state the proof for $\tau = 1$ by noting it follows similar arguments as for $\tau = 2$. Let $\tilde{u}$ be a solution of (2.1). Note, that there exists $u \in BV(\Omega)$ such that $\tilde{u} = u + \frac{1}{|\Omega|}\int_\Omega g$. We consider now the continuous function $f(s) = \|T(su + \frac{1}{|\Omega|}\int_\Omega g) - g\|_{L^1(\Omega)}$ for $s \in [0, 1]$. Note that $f(1) = \|T\tilde{u} - g\|_{L^1(\Omega)} \leq \nu_1|\Omega|$ and $f(0) = \|T(\frac{1}{|\Omega|}\int_\Omega g) - g\|_{L^1(\Omega)} = \|g - \frac{1}{|\Omega|}\int_\Omega g\|_{L^1(\Omega)} \geq \nu_1|\Omega|$, since $T \cdot 1 = 1$, and hence there exists some $s \in [0, 1]$ such that $f(s) = \nu_1|\Omega|$. Set $u' = su$ which satisfies $\|Tu' - g\|_{L^1(\Omega)} = \nu_1|\Omega|$ and

$$\mathcal{R}(u') = s\mathcal{R}(u) \leq \liminf_{k\to\infty} \mathcal{R}(u_{n_k}).$$

Hence u' is a solution of (1.8). □

Now we are able to argue the equivalence of the problems (1.6) and (1.8).

Theorem 2.2 *Let $T \in \mathcal{L}(L^2(\Omega))$ be such that $T \cdot 1 = 1$ and $\nu_\tau|\Omega| \leq \|g - \frac{1}{|\Omega|}\int_\Omega g\|^\tau_{L^\tau(\Omega)}$. Then there exists $\alpha \geq 0$ such that the constrained minimization problem* (1.8) *is equivalent to the unconstrained problem* (1.6)*, i.e., u is a solution of* (1.8) *if and only if u solves* (1.6)*.*

Proof For $\tau = 2$ the proof can be found in [18, Prop. 2.1]. By similar arguments one can show the statement for $\tau = 1$, which we state here.

Set

$$G(u) = \begin{cases} +\infty & \text{if } \|u - g\|_{L^1(\Omega)} > \nu_1|\Omega|, \\ 0 & \text{if } \|u - g\|_{L^1(\Omega)} \leq \nu_1|\Omega|. \end{cases}$$

Notice that $\mathcal{R}$ and G are convex l.s.c functions and problem (2.1) is equivalent to $\min_u \mathcal{R}(u) + G(Tu)$. We have $\mathrm{Dom}(\mathcal{R}) = BV(\Omega) \cap L^2(\Omega)$ and $\mathrm{Dom}(G) = \{u \in L^2(\Omega) : G(u) < +\infty\}$. Since $g \in \overline{T\,\mathrm{Dom}(\mathcal{R})}$, there exists $\tilde{u} \in \mathrm{Dom}(\mathcal{R})$ with $\|T\tilde{u} - g\|_{L^1(\Omega)} \leq \nu_1|\Omega|/2$. As $T \in \mathcal{L}(L^2(\Omega))$ is continuous, $G \circ T$ is continuous at $\tilde{u}$. Hence, by [33, Prop. 5.6, p. 26] we obtain

$$\partial(\mathcal{R} + G \circ T)(u) = \partial\mathcal{R}(u) + \partial(G \circ T)(u)$$

for all u. Further, G is continuous at $T\tilde{u}$, and hence by [33, Prop. 5.7, p. 27] we have for all u,

$$\partial(G \circ T)(u) = T^*\partial G(Tu)$$

where $\partial G(u) = \{0\}$ if $\|u - g\|_{L^1(\Omega)} < \nu_1|\Omega|$ and $\partial G(u) = \{\alpha\partial(\|u - g\|_{L^1(\Omega)}), \alpha \geq 0\}$ if $\|u - g\|_{L^1(\Omega)} = \nu_1|\Omega|$.

If u is a solution of (2.1) and hence of (1.8), then

$$0 \in \partial(\mathcal{R} + G \circ T)(u) = \partial\mathcal{R}(u) + T^*\partial G(Tu).$$

Since any solution of (1.8) satisfies $\|Tu - g\|_{L^1(\Omega)} = \nu_1|\Omega|$, this shows that there exists an $\alpha \geq 0$ such that

$$0 \in \partial\mathcal{R}(u) + \alpha T^*\partial(\|Tu - g\|_{L^1(\Omega)}).$$

Hence for this $\alpha \geq 0$, u is a minimizer of the problem in (1.6).

Conversely, a minimizer u of (1.6) with the above α is obviously a solution of (1.8) with $\|Tu - g\|_{L^1(\Omega)} = \nu_1|\Omega|$. This concludes the proof. □

Note that $\|Tu_\alpha - g\|_{L^1(\Omega)}$ is (only) convex with respect to Tu_α, and hence Tu_α is in general not unique. On the contrary, $\|Tu_\alpha - g\|^2_{L^2(\Omega)}$ is strictly convex with respect to Tu_α. This concludes that $\alpha \mapsto \mathcal{H}_1(u_\alpha; g)$ is in general not continuous, while $\alpha \mapsto \mathcal{H}_2(u_\alpha; g)$ indeed is continuous, where u_α is a respective minimizer of (1.6). We have the following properties:

Lemma 2.1 *Let $u_\alpha \in \arg\min_{u\in BV(\Omega)} \mathcal{H}_\tau(u; g) + \alpha\mathcal{R}(u)$, then the function $\alpha \mapsto \mathcal{H}_\tau(u_\alpha; g)$ is non-decreasing for $\tau = 1, 2$. Moreover, $\alpha \mapsto \mathcal{H}_2(u_\alpha; g)$ maps $\mathbb{R}^+$ onto $[0, \|g - \frac{1}{|\Omega|}\int_\Omega g\|^2_{L^2(\Omega)}]$.*

Proof For a proof see [18,20]. □

Proposition 2.2 *If u_{α_i} is a minimizer of*

$$\mathcal{E}(u, \alpha_i) := \|u - g\|^2_{L^2(\Omega)} + \alpha_i\mathcal{R}(u)$$

for $i = 1, 2$, then we have

$$\|u_{\alpha_1} - u_{\alpha_2}\|_{L^2(\Omega)} \leq C \left\| g - \frac{1}{|\Omega|}\int_\Omega g \right\|_{L^2(\Omega)}.$$

with $C := \min\left\{2\left|\frac{\alpha_2 - \alpha_1}{\alpha_2 + \alpha_1}\right|, \left|\frac{\alpha_2 - \alpha_1}{\alpha_2 + \alpha_1}\right|^{\frac{1}{2}}\right\}$

Proof By [7, Lemma 10.2] we have

$$\frac{1}{\alpha_1}\|u_{\alpha_1} - u_{\alpha_2}\|^2_{L^2(\Omega)} \leq \frac{1}{\alpha_1}\left(\mathcal{E}(u_{\alpha_2}, \alpha_1) - \mathcal{E}(u_{\alpha_1}, \alpha_1)\right)$$
$$\frac{1}{\alpha_2}\|u_{\alpha_1} - u_{\alpha_2}\|^2_{L^2(\Omega)} \leq \frac{1}{\alpha_2}\left(\mathcal{E}(u_{\alpha_1}, \alpha_2) - \mathcal{E}(u_{\alpha_2}, \alpha_2)\right).$$

Summing up these two inequalities yields

$$\left(\frac{1}{\alpha_1} + \frac{1}{\alpha_2}\right)\|u_{\alpha_1} - u_{\alpha_2}\|^2_{L^2(\Omega)}$$
$$\leq \left(\frac{1}{\alpha_1} - \frac{1}{\alpha_2}\right)\left(\|u_{\alpha_2} - g\|^2_{L^2(\Omega)} - \|u_{\alpha_1} - g\|^2_{L^2(\Omega)}\right)$$

which implies

$$\|u_{\alpha_1} - u_{\alpha_2}\|^2_{L^2(\Omega)} \leq \frac{\alpha_2 - \alpha_1}{\alpha_1 + \alpha_2} \left(\|u_{\alpha_2} - g\|^2_{L^2(\Omega)} - \|u_{\alpha_1} - g\|^2_{L^2(\Omega)} \right) \tag{2.2}$$

By the non-decrease and boundedness of the function $\alpha \mapsto \mathcal{H}_2(u_\alpha; g)$, see Lemma 2.1, it follows

$$\|u_{\alpha_1} - u_{\alpha_2}\|^2_{L^2(\Omega)} \leq \left| \frac{\alpha_2 - \alpha_1}{\alpha_1 + \alpha_2} \right| \left\| g - \frac{1}{|\Omega|} \int_\Omega g \right\|^2_{L^2(\Omega)}. \tag{2.3}$$

On the other hand inequality (2.2) implies

$$\|u_{\alpha_1} - u_{\alpha_2}\|_{L^2(\Omega)} \leq \left| \frac{\alpha_2 - \alpha_1}{\alpha_1 + \alpha_2} \right| \left(\|u_{\alpha_2} - g\|_{L^2(\Omega)} + \|u_{\alpha_1} - g\|_{L^2(\Omega)} \right),$$

where we used the binomial formula $a^2 - b^2 = (a + b)(a - b)$ for $a, b \in \mathbb{R}$ and the triangle inequality. Using Lemma 2.1 yields

$$\|u_{\alpha_1} - u_{\alpha_2}\|_{L^2(\Omega)} \leq 2 \left| \frac{\alpha_2 - \alpha_1}{\alpha_1 + \alpha_2} \right| \left\| g - \frac{1}{|\Omega|} \int_\Omega g \right\|_{L^2(\Omega)}. \tag{2.4}$$

The assertion follows then from (2.3) and (2.4). □

Remark 2.1 Without loss of generality let $\alpha_2 \geq \alpha_1$ in Proposition 2.2, then we easily check that

$$C = \begin{cases} \left(\frac{\alpha_2 - \alpha_1}{\alpha_2 + \alpha_1} \right)^{\frac{1}{2}} & \text{if } \alpha_2 > \frac{5}{3}\alpha_1, \\ \frac{1}{2} & \text{if } \alpha_2 = \frac{5}{3}\alpha_1, \\ 2\frac{\alpha_2 - \alpha_1}{\alpha_2 + \alpha_1} & \text{otherwise.} \end{cases}$$

3 Notations and Basic Definitions

In the rest of the paper we consider a discretization of the continuous image domain $\Omega \subset \mathbb{R}^2$, again denoted by Ω. That is, we work on a finite regular mesh of size $N_1 \times N_2$, where $N_1, N_2 \in \mathbb{N}$, with equidistant mesh size $h = x_{i+1,j} - x_{i,j} = x_{i,j+1} - x_{i,j}$ equal to 1 (one pixel). By $|\Omega| = N_1 N_2$ we denote the size of the discrete image (number of pixels). We approximate functions u by discrete functions, again denoted by u. The considered function spaces are $X = \mathbb{R}^{N_1 \times N_2}$ and $Y = X \times X$ with the associated scalar products

$$\langle u, v \rangle_X = \sum_{x \in \Omega} u(x) v(x)$$

for $u, v \in X$ and

$$\langle \mathbf{p}, \mathbf{q} \rangle_Y = \sum_{x \in \Omega} \langle \mathbf{p}(x), \mathbf{q}(x) \rangle_{\mathbb{R}^2}$$

for $\mathbf{p}, \mathbf{q} \in Y$ with $\langle y, z \rangle_{\mathbb{R}^2} = \sum_{j=1}^2 y_j z_j$ for every $y = (y_1, y_2) \in \mathbb{R}^2$ and $z = (z_1, z_2) \in \mathbb{R}^2$. In what follows for all $u \in X$ we use the following norms

$$\|u\|_\tau := \|u\|_{\ell^\tau(\Omega)} = \left(\sum_{x \in \Omega} |u(x)|^\tau \right)^{1/\tau}$$

for $1 \leq \tau < +\infty$. Moreover we denote by $\langle u \rangle$ the average value of $u \in X$, i.e, $\langle u \rangle := \frac{1}{|\Omega|} \sum_{x \in \Omega} u(x)$. The discrete gradient ∇u is a vector in $Y = X \times X$ given by

$$(\nabla u)(x) = ((\nabla u)^1(x), (\nabla u)^2(x)),$$

where

$$(\nabla u)^1(x_{i,j}) = \begin{cases} u(x_{i+1,j}) - u(x_{i,j}) & \text{if } i < N_1, \\ 0 & \text{if } i = N_1, \end{cases}$$

and

$$(\nabla u)^2(x_{i,j}) = \begin{cases} u(x_{i,j+1}) - u(x_{i,j}) & \text{if } j < N_2, \\ 0 & \text{if } j = N_2, \end{cases}$$

for all $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$. By T^* we denote the adjoint (or transpose) of an operator T. Further we introduce the *discrete divergence* $\operatorname{div} : Y \to X$ as $\operatorname{div} := -\nabla^*$ (∇^* is the adjoint of the gradient ∇), in analogy to the continuous setting. In our case, the discrete divergence operator is explicitly given by

$$(\operatorname{div} \mathbf{p})(x_{i,j}) = \begin{cases} p^1(x_{i,j}) - p^1(x_{i-1,j}) & \text{if } 1 < i < N_1, \\ p^1(x_{i,j}) & \text{if } i = 1, \\ -p^1(x_{i-1,j}) & \text{if } i = N_1, \end{cases} + \begin{cases} p^2(x_{i,j}) - p^2(x_{i,j-1}) & \text{if } 1 < j < N_2, \\ p^2(x_{i,j}) & \text{if } j = 1, \\ -p^2(x_{i,j-1}) & \text{if } j = N_2, \end{cases}$$

for every $\mathbf{p} = (p^1, p^2) \in Y$. We define the closed convex set

$$K = \{ \operatorname{div} \mathbf{p} : \mathbf{p} \in Y, |\mathbf{p}(x)|_{l^2} \leq 1 \ \forall x \in \Omega \},$$

where $|y|_{l^2} = \sqrt{y_1^2 + y_2^2}$ for every $y = (y_1, y_2) \in \mathbb{R}^2$, and denote $P_K(u) := \arg\min_{v \in K} \|u - v\|_2$ the orthogonal projection onto K.

With the above notations and definitions the discretization of the general function in (1.9) is given by

$$E_\tau(u, \alpha) := H_\tau(u) + R_\alpha(u)$$

where $H_\tau(u) = \frac{1}{\tau}\|Tu-g\|^\tau_{\ell^\tau(\Omega)}$, $\tau \in \{1,2\}$, $T \in \mathcal{L}(X)$ is a bounded linear operator, $\alpha \in (\mathbb{R}^+)^{N_1\times N_2}$, and

$$R_\alpha(u) := \sum_{x\in\Omega} \alpha(x)|\nabla u(x)|_{l^2}. \tag{3.1}$$

In the sequel if α is a scalar or $\alpha \equiv 1$ in (3.1), we write instead of R_α or R_1 just αR or R, respectively, i.e.,

$$R(u) = \sum_{x\in\Omega} |\nabla u(x)|_{l^2}$$

is the discrete total variation of u in Ω. Introducing some step-size h, then for $h \to 0$ (i.e. the number of pixels N_1N_2 goes to infinity) one can show, similar as for the case $\alpha \equiv 1$, that R_α Γ-converges to $\int_\Omega \alpha|Du|$; see [12,53]. Throughout the paper we assume that $T \cdot 1 = 1$ (i.e., T does not annihilate constant functions).

4 Automated parameter selection

In order to find a suitable regularization parameter $\alpha > 0$ of the minimization problem

$$\min_{u\in X}\{\bar{E}_\tau(u,\alpha) := H_\tau(u) + \alpha R(u)\}, \tag{4.1}$$

we consider the corresponding constrained optimization problem

$$\min_{u\in X} R(u) \quad \text{s.t.} \quad H_\tau(u) \le \frac{\nu_\tau}{\tau}|\Omega| =: B_\tau \tag{4.2}$$

where $\nu_\tau \in \mathbb{R}$ is a statistical value depending on the underlying noise and possibly on the original image.

Since T does not annihilate constant functions the existence of a solution of the constrained minimization problem (4.2) is guaranteed; see Section 2. Moreover, by Proposition 2.1 we have that problem (4.2) is equivalent to the constrained minimization problem

$$\min_{u\in X} R(u) \quad \text{s.t.} \quad H_\tau(u) = B_\tau. \tag{4.3}$$

4.1 Statistical characterization of the noise

Let us characterize the noise corrupting the image in more details by making similar considerations as in [50, Section 2]. Note, that at any point $x \in \Omega$ the contaminated image $g(x) = \mathcal{N}(T\hat{u})(x)$ is a stochastic observation, which depends on the underlying noise. Two important measures to characterize noise are the expected absolute value and the variance which we denote by ν_1 and ν_2 respectively. For images contaminated by Gaussian white noise with standard deviation σ, we typically set $\tau = 2$ and $\nu_2 = \sigma^2$. If the image is instead corrupted by impulse noise, then we set $\tau = 1$ and we have to choose ν_1 properly. In particular, for salt-and-pepper noise $\nu_1 \in [\min\{r_1,r_2\}, \max\{r_1,r_2\}]$, while for random-valued impulse noise ν_1 should be a value in the interval $[\frac{r}{4}, \frac{r}{2}]$, where we used that for any point $x \in \Omega$ we have $T\hat{u}(x) \in [0,1]$; cf. [50]. Here ν_1 seems to be fixed, while actually ν_1 depends on the true (unknown) image $\hat{u}$. In particular, for salt-and-pepper noise the expected absolute value is given by

$$\nu_1(\hat{u}) := r_1 - (r_1 - r_2)\frac{1}{|\Omega|}\sum_{x\in\Omega}(T\hat{u})(x) \tag{4.4}$$

and for random-valued impulse noise we have

$$\nu_1(\hat{u}) := \frac{1}{|\Omega|}\sum_{x\in\Omega} r\left((T\hat{u})(x)^2 - (T\hat{u})(x) + \frac{1}{2}\right). \tag{4.5}$$

However, instead of considering the constraint $H_\tau(u) = B_\tau(u)$ in (4.3) which results in a quite nonlinear problem, in our numerics we choose a reference image and compute an approximate value B_τ. Note that for salt-and-pepper noise with $r_1 = r_2$ the expected absolute value becomes independent of $\hat{u}$ and hence $\nu_1 = r_1$. In case of Gaussian noise ν_τ and B_τ are independent of $\hat{u}$ too. Nevertheless, in order to keep the paper concise, in the sequel instead of ν_τ and B_τ we often write $\nu_\tau(\tilde{u})$ and $B_\tau(\tilde{u})$, where $\tilde{u}$ represents a reference image approximating $\hat{u}$, even if the values may actually be independent from the image.

4.2 Automated parameter selection

In order to determine a suitable regularization parameter α in [16] an algorithm for solving the constrained minimization problem (4.3) for $T = I$ and $\tau = 2$ is proposed, i.e., in the presence of Gaussian noise with zero mean and standard deviation σ. This algorithm relies on the fact that $\alpha \mapsto H_2(u_\alpha)$ is non-decreasing, which leads to the following iterative procedure.

Chambolle's parameter selection (CPS):
Choose $\alpha_0 > 0$ and set $n := 0$.

1) Compute $u_{\alpha_n} \in \arg\min_{u\in X} \|u-g\|_2^2 + 2\alpha_n R(u)$
2) Update $\alpha_{n+1} := \frac{\sigma\sqrt{|\Omega|}}{\|u_{\alpha_n}-g\|_2}\alpha_n$.
3) Stop or set $n := n+1$ and return to step 1).

For the minimization of the optimization problem in step 1) in [16] a method based on the dual formulation of the total variation is used. However, we note that any other algorithm for total variation minimization might be used for solving this minimization problem. The CPS-algorithm generates a sequence $(u_{\alpha_n})_n$ such that for $n \to \infty$, $\|u_{\alpha_n} - g\|_2 \to \sigma\sqrt{|\Omega|}$ and u_{α_n} converges to the unique solution of (4.2) with $T = I$ and

$\tau = 2$ [16]. The proof relies on the fact that the function $\alpha \to \frac{\|u_\alpha - g\|_2}{\alpha}$ is non-increasing. Note, that this property does not hold in general for operators $T \neq I$.

Due to the monotonic behavior of the function $\alpha \mapsto H_\tau(u_\alpha)$, see Lemma 2.1, we generalize the CPS-algorithm to optimization problems of the type (4.2) for $\tau = 1, 2$ and for general operators T. An ad hoc generalization leads to the following automated parameter selection algorithm.

APS-algorithm: Choose $\alpha_0 > 0$ and set $n := 0$.

1) Compute $u_{\alpha_n} \in \arg\min_{u \in X} \bar{E}_\tau(u, \alpha_n)$
2) Update $\alpha_{n+1} := \frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\alpha_n$ if $H_\tau(u_{\alpha_n}) > 0$ and continue with step 3). Otherwise increase α_n, e.g., $\alpha_n := 10\alpha_n$, and go to step 1).
3) Stop or set $n := n + 1$ and return to step 1).

Under certain assumptions we can immediately argue the convergence of the APS-algorithm.

Theorem 4.1 *For $\alpha > 0$ let $u_\alpha \in \arg\min_{u \in X} \bar{E}_\tau(u, \alpha)$. Assume that $B_\tau(u) \equiv B_\tau$ is a constant independent of u, the function $\alpha \mapsto \frac{H_\tau(u_\alpha)}{\alpha}$ is non-increasing, and $\|g - \langle g \rangle\|_\tau^\tau \geq \nu_\tau |\Omega|$. Then the APS-algorithm generates a sequence $(\alpha_n)_n \subset \mathbb{R}^+$ such that $\lim_{n\to\infty} \alpha_n = \bar{\alpha} > 0$, $\lim_{n\to\infty} H_\tau(u_{\alpha_n}) = B_\tau$ and u_{α_n} converges to $u_{\bar{\alpha}} \in \arg\min_{u \in X} H_\tau(u) + \bar{\alpha} R(u)$ for $n \to \infty$.*

Proof Let us start with assuming that $H_\tau(u_{\alpha_0}) < B_\tau$. By induction, we can show that $\alpha_n < \alpha_{n+1}$ unless $H_\tau(u_{\alpha_n}) = B_\tau$ and $H_\tau(u_{\alpha_n}) \leq H_\tau(u_{\alpha_{n+1}}) \leq B_\tau$. In particular, if $H_\tau(u_{\alpha_n}) < B_\tau$ then $\alpha_{n+1} = \frac{B_\tau}{H_\tau(u_{\alpha_n})}\alpha_n > \alpha_n$ and by Lemma 2.1 it follows that $H_\tau(u_{\alpha_n}) \leq H_\tau(u_{\alpha_{n+1}})$. Moreover, by the assumption that $\alpha \mapsto \frac{H_\tau(u_\alpha)}{\alpha}$ is non-increasing we obtain $H_\tau(u_{\alpha_{n+1}}) \leq \frac{\alpha_{n+1}}{\alpha_n} H_\tau(u_{\alpha_n}) = B_\tau$. That is,

$$H_\tau(u_{\alpha_n}) \leq H_\tau(u_{\alpha_{n+1}}) \leq B_\tau.$$

Note, that there exists an $\alpha^* > 0$ with $H_\tau(u_{\alpha^*}) = B_\tau$, see Theorem 2.2, such that for any $\alpha \geq \alpha^*$, $H_\tau(u_\alpha) \geq B_\tau$; cf. Lemma 2.1. If $\alpha_n \geq \alpha^*$, then $H_\tau(u_{\alpha_n}) \geq B_\tau$. Hence $H_\tau(u_{\alpha_n}) = B_\tau$ and $\alpha_{n+1} = \alpha_n$. Thus we deduce that the sequences $(H_\tau(u_{\alpha_n}))_n$ and $(\alpha_n)_n$ are non-decreasing and bounded. Consequently, there exists an $\bar{\alpha}$ such that $\lim_{n\to\infty} \alpha_n = \bar{\alpha}$ with $H_\tau(u_{\bar{\alpha}}) = B_\tau$. Let $\bar{H} = \lim_{n\to\infty} H_\tau(u_{\alpha_n})$, then $\bar{H} = H_\tau(u_{\bar{\alpha}}) = B_\tau$. By the optimality of u_{α_n} we have that $0 \in \partial \bar{E}_\tau(u_{\alpha_n}, \alpha_n) = \partial H_\tau(u_{\alpha_n}) + \alpha_n \partial R(u_{\alpha_n})$; see [33, Prop. 5.6 + Eq. (5.21), p.26]. Consequently there exist $v_{\alpha_n} \in \partial R(u_{\alpha_n})$ such that $-\alpha_n v_{\alpha_n} \in \partial H_\tau(u_{\alpha_n})$ with $\lim_{n\to\infty} v_{\alpha_n} = v_{\bar{\alpha}}$. By [66, Thm. 24.4, p. 233] we obtain that $-\bar{\alpha} v_{\bar{\alpha}} \in \partial H_\tau(u_{\bar{\alpha}})$ with $v_{\bar{\alpha}} \in \partial R(u_{\bar{\alpha}})$ and hence $0 \in \partial \bar{E}_\tau(u_{\bar{\alpha}}, \bar{\alpha})$ for $n \to \infty$.

If $H_\tau(u_{\alpha_0}) > B_\tau$, then as above we can show by induction that $\alpha_n > \alpha_{n+1}$ unless $H_\tau(u_{\alpha_n}) = B_\tau$ and $H_\tau(u_{\alpha_n}) \geq H_\tau(u_{\alpha_{n+1}}) \geq B_\tau$. Thus we deduce that $(H_\tau(u_{\alpha_n}))_n$ and $(\alpha_n)_n$ are non-increasing and bounded. Note, that there exists an $\alpha^* > 0$ with $H_\tau(u_{\alpha^*}) = B_\tau$ such that for any $\alpha \leq \alpha^*$, $H_\tau(u_\alpha) \leq B_\tau$. Hence if $\alpha_n \leq \alpha^*$, then $H_\tau(u_{\alpha_n}) \leq B_\tau$. Hence $H_\tau(u_{\alpha_n}) = B_\tau$ and $\alpha_{n+1} = \alpha_n$. The rest of the proof is identical to above.

If $H_\tau(u_{\alpha_0}) = B_\tau$ the assertion follows immediately. □

Nothing is known about the convergence of the APS-algorithm, if $B_\tau(\cdot)$ indeed depends on u and $B_\tau(u_{\alpha_n})$ is used instead of a fixed constant. In particular, in our numerics for some examples we even observe that starting from a certain iteration the sequence $(\alpha_n)_n$ oscillates between two states, see Figure 7.3. This behavior can be attributed to the fact that, for example, if $H_\tau(u_{\alpha_n}) \leq B_\tau(u_{\alpha_n})$, then it is not guaranteed that also $H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_{n+1}})$, which is essential for the convergence.

The second assumption in the previous theorem, i.e., the non-increase of the function $\alpha \mapsto \frac{H_\tau(u_\alpha)}{\alpha}$, can be slightly loosened, since for the convergence of the APS-algorithm it is enough to demand the non-increase starting from a certain iteration $\tilde{n} \geq 0$. That is, if there exists a region $U \subset \mathbb{R}^+$ where $\alpha \mapsto \frac{H_\tau(u_\alpha)}{\alpha}$ is non-increasing and $(\alpha_n)_{n \geq \tilde{n}} \subset U$, then the algorithm converges; see Figure 7.2. Analytically, this can be easily shown via Theorem 4.1 by just considering $\alpha_{\tilde{n}}$ as the initial value of the algorithm. However, if $\tau = 2$, similar to the CPS-algorithm, we are able to show the following monotonicity property.

Proposition 4.1 *Let $u_\alpha \in \arg\min_{u \in X} \bar{E}_2(u, \alpha)$. If there exists a constant $c > 0$ such that $\|T^*(Tu - g)\|_2 = c\|Tu - g\|_2$ for all $u \in X$, then the function $\alpha \mapsto \frac{\sqrt{H_2(u_\alpha)}}{\alpha}$ is non-increasing.*

Proof We start by replacing the functional $\bar{E}_2$ by the family of surrogate functionals denoted by $\bar{\mathcal{S}}$ and defined for $u, a \in X$ as

$$\begin{aligned}\bar{\mathcal{S}}(u, a) &:= \bar{E}_2(u, \alpha) + \frac{\delta}{2}\|u - a\|_2^2 - \frac{1}{2}\|T(u - a)\|_2^2 \\ &= \delta\|u - z(a)\|_2^2 + 2\alpha R(u) + \psi(a, g, T)\end{aligned}$$

where $\delta > \|T\|^2$, $z(a) := a - \frac{1}{\delta}T^*(Ta - g)$, and ψ is a function independent of u. It can be shown that the iteration

$$u_{\alpha,0} \in X, \quad u_{\alpha,k+1} = \arg\min_u \bar{\mathcal{S}}(u, u_{\alpha,k}), \ k \geq 0 \qquad (4.6)$$

generates a sequence $(u_{\alpha,k})_k$ which converges for $k \to \infty$ to a minimizer u_α of $\bar{E}_2(u, \alpha)$, see for example [28]. The unique minimizer $u_{\alpha,k+1}$ is given by $u_{\alpha,k+1} = (I -$

$P_{\frac{\alpha}{\delta}K})(z(u_{\alpha,k}))$; see [16]. Then for $k \to \infty$, let us define $\tilde{f}(\alpha) := \|P_{\frac{\alpha}{\delta}K}(z(u_\alpha))\|_2 = \|\frac{1}{\delta}T^*(Tu_\alpha - g)\|_2$. Since $\|T^*(Tu-g)\|_2 = c\|Tu-g\|_2$, it follows that $\tilde{f}(\alpha) = \frac{c}{\delta}\sqrt{2H_2(u_\alpha)}$. The assertion follows by applying [16, Lemma 4.1] to $\tilde{f}$ and by noting that the non-increase of $\alpha \mapsto \frac{\tilde{f}(\alpha)}{\alpha}$ implies the non-increase of $\alpha \mapsto \frac{\sqrt{H_2(u_\alpha)}}{\alpha}$. □

We remark that for convolution type of operators the assumption of Proposition 4.1 does not hold in general. However, there exist several operators T, relevant in image processing, with the property $\|T^*(Tu-g)\|_2 = \|Tu-g\|_2$. Such operators include $T = I$ for image denoising, $T = 1_D$ for image inpainting, and $T = S \circ A$, where S is a subsampling operator and A is an analysis operator of a Fourier or orthogonal wavelet transform. The latter type of operator is used for reconstructing signals from partial Fourier data [15] or in wavelet inpainting [22], respectively. For all such operators the function $\alpha \mapsto \frac{\sqrt{H_2(u_\alpha)}}{\alpha}$ is non-increasing and hence by changing the update of α in the APS-algorithm to

$$\alpha_{n+1} := \sqrt{\frac{B_2}{H_2(u_{\alpha_n})}}\alpha_n,$$

where B_2 is a fixed constant, we obtain in these situations a convergent algorithm. We emphasize, that in general the non-increase of the function $\alpha \mapsto \frac{H_\tau(u_\alpha)}{\alpha}$ is not guaranteed. Nevertheless, there exists always a constant $p \geq 0$ such that $\alpha \mapsto \frac{(H_\tau(u_\alpha))^p}{\alpha}$ is indeed non-increasing. For example, $p = \frac{1}{2}$ for operators T with the property $\|T^*(Tu-g)\|_2 = \|Tu-g\|_2$; cf. Propsition 4.1. In particular, one easily checks the following result.

Proposition 4.2 *Let $0 < \alpha \leq \beta$, and u_α and u_β minimizers of $\bar{E}_\tau(\cdot,\alpha)$ and $\bar{E}_\tau(\cdot,\beta)$, respectively, for $\tau = 1, 2$. Then $\frac{(H_\tau(u_\beta))^p}{\beta} \leq \frac{(H_\tau(u_\alpha))^p}{\alpha}$ if and only if $p \leq \frac{\ln\beta - \ln\alpha}{\ln H_\tau(u_\beta) - \ln H_\tau(u_\alpha)}$.*

In fact, we desire for a $p > 0$ such that in every iteration $n \in \mathbb{N}$ we have

$$\frac{(H_\tau(u_{\alpha_{n+1}}))^p}{\alpha_{n+1}} \leq \frac{(H_\tau(u_{\alpha_n}))^p}{\alpha_n} \tag{4.7}$$

if $\alpha_{n+1} \geq \alpha_n$ or

$$\frac{(H_\tau(u_{\alpha_{n+1}}))^p}{\alpha_{n+1}} \geq \frac{(H_\tau(u_{\alpha_n}))^p}{\alpha_n} \tag{4.8}$$

if $\alpha_{n+1} < \alpha_n$. Therefore, for a fixed α_n the update of α has to be adjusted to

$$\alpha_{n+1} := \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n. \tag{4.9}$$

Proposition 4.3 *Let $\alpha_n > 0$ be a fixed constant and $u_{\alpha_n} \in \arg\min_{u\in X} \bar{E}_\tau(u,\alpha_n)$.*

(i) If $p \geq 0$ such that $0 < H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_n})$ with α_{n+1} as in (4.9), then $\frac{(H_\tau(u_{\alpha_{n+1}}))^p}{\alpha_{n+1}} \leq \frac{(H_\tau(u_{\alpha_n}))^p}{\alpha_n}$.

(ii) If $p \geq 0$ such that $H_\tau(u_{\alpha_{n+1}}) \geq B_\tau(u_{\alpha_n})$ with α_{n+1} as in (4.9), then $\frac{(H_\tau(u_{\alpha_{n+1}}))^p}{\alpha_{n+1}} \geq \frac{(H_\tau(u_{\alpha_n}))^p}{\alpha_n}$.

Proof Since H_τ is bounded from below by zero and $B_\tau > 0$, we have

$$\begin{aligned}
&H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_n}) \Leftrightarrow 0 < \frac{H_\tau(u_{\alpha_{n+1}})}{B_\tau(u_{\alpha_n})} \leq 1\\
&\quad\Leftrightarrow \ln\left(\frac{H_\tau(u_{\alpha_{n+1}})}{B_\tau(u_{\alpha_n})}\right) \leq 0 \Leftrightarrow p\ln\left(\frac{H_\tau(u_{\alpha_{n+1}})}{B_\tau(u_{\alpha_n})}\right) \leq 0\\
&\quad\Leftrightarrow \left(\frac{H_\tau(u_{\alpha_{n+1}})}{B_\tau(u_{\alpha_n})}\right)^p \leq 1\\
&\quad\Leftrightarrow \frac{\left(H_\tau(u_{\alpha_{n+1}})\right)^p}{\left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n}\frac{\alpha_n}{H_\tau(u_{\alpha_n})^p} \leq 1,
\end{aligned}$$

which shows the first assertion. The second statement can be shown analogously. □

From the above proof we see that the converse direction of the statement also holds true.

Moreover, the existence of $p > 0$ such that the above conditions hold for any $n \in \mathbb{N}$ is guaranteed by the following statement.

Proposition 4.4 *Assume $\|g - \langle g\rangle\|_\tau^\tau \geq \nu_\tau|\Omega|$ and α_{n+1} is defined as in (4.9).*

(i) If $\alpha_n > 0$ such that $H_\tau(u_{\alpha_n}) = B_\tau(u_{\alpha_n})$, then for all $p \in \mathbb{R}$ we have $H_\tau(u_{\alpha_{n+1}}) = B_\tau(u_{\alpha_n})$.

(ii) If $\alpha_n > 0$ such that $0 < H_\tau(u_{\alpha_n}) < B_\tau(u_{\alpha_n})$, then there exist $p > 0$ with $H_\tau(u_{\alpha_n}) \leq H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_n})$.

(iii) If $\alpha_n > 0$ such that $H_\tau(u_{\alpha_n}) > B_\tau(u_{\alpha_n})$, then there exist $p > 0$ with $H_\tau(u_{\alpha_n}) \geq H_\tau(u_{\alpha_{n+1}}) \geq B_\tau(u_{\alpha_n})$.

Proof (i) Since $H_\tau(u_{\alpha_n}) = B_\tau(u_{\alpha_n})$, it follows that $\alpha_{n+1} = \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n = \alpha_n$ and thus $H_\tau(u_{\alpha_{n+1}}) = B_\tau(u_{\alpha_n})$ for all $p \in \mathbb{R}$.

(ii) Now let $0 < H_\tau(u_{\alpha_n}) < B_\tau(u_{\alpha_n})$, then we have $\alpha_{n+1} = \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n > \alpha_n$ for all $p > 0$. By Theorem 2.2 there exists an $\alpha^* \geq 0$ such that $H_\tau(u_{\alpha^*}) = B_\tau(u_{\alpha_n})$ and since $\alpha \mapsto H_\tau(u_\alpha)$ is non-decreasing, we have $\alpha^* > \alpha_n$. Further, since $\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})} > 1$ and by the continuity of α_{n+1} with respect to p, there exists a $p^* > 0$ such that $\alpha^* = \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^{p^*} \alpha_n$. For any positive $p \leq p^*$ the assertion follows by setting $\alpha_{n+1} = \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n$.

(iii) From $H_\tau(u_{\alpha_n}) > B_\tau(u_{\alpha_n})$ it follows that $\alpha_{n+1} = \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n < \alpha_n$ for any $p > 0$. Since $\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})} < 1$ and by the same arguments as above there exists $p^* > 0$ such that $H_\tau(u_{\alpha_{n+1}}) = B_\tau(u_{\alpha_n})$ with $\alpha_{n+1} = \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^{p^*} \alpha_n < \alpha_n$. The assertion follows for any positive $p \leq p^*$.

□

If $B_\tau(u_{\alpha_n}) \equiv B_\tau$ is a constant independent of u_{α_n}, which is the case for Gaussian noise removal, i.e., $\tau = 2$, then Proposition 4.4 together with Proposition 4.3 says that the conditions in (4.7) and (4.8) are sufficient to guarantee that either $H_\tau(u_{\alpha_n}) \leq B_\tau$ or $H_\tau(u_{\alpha_n}) \geq B_\tau$ for all $n \geq 0$, which prevents the sequence $(\alpha_n)_n$ to oscillate. However, if B_τ indeed depends on u_{α_n}, then the conditions (4.7) and (4.8) do not ensure that $H_\tau(u_{\alpha_n}) \leq B_\tau(u_{\alpha_n})$ and $H_\tau(u_{\alpha_n}) \geq B_\tau(u_{\alpha_n})$ for all n, respectively. Hence it is preferable to use instead directly the conditions $H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_{n+1}})$ and $H_\tau(u_{\alpha_{n+1}}) \geq B_\tau(u_{\alpha_{n+1}})$. For these latter two conditions only the existence of a $p \geq 0$ for every n can be argued.

Proposition 4.5 *Assume $\|g - \langle g\rangle\|_\tau^\tau \geq \nu_\tau |\Omega|$ and α_{n+1} is defined as in* (4.9).

(i) If $\alpha_n > 0$ such that $H_\tau(u_{\alpha_n}) = B_\tau(u_{\alpha_n})$, then for all $p \in \mathbb{R}$ we have that $H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_{n+1}})$.
(ii) If $\alpha_n > 0$ such that $0 < H_\tau(u_{\alpha_n}) < B_\tau(u_{\alpha_n})$, then there exist $p \geq 0$ with $H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_{n+1}})$.
(iii) If $\alpha_n > 0$ such that $H_\tau(u_{\alpha_n}) > B_\tau(u_{\alpha_n})$, then there exist $p \geq 0$ with $H_\tau(u_{\alpha_{n+1}}) \geq B_\tau(u_{\alpha_{n+1}})$.

Proof The assertion immediately follows by noting that for $p = 0$ we have $\alpha_{n+1} = \alpha_n$. □

Moreover, due to the dependency of α_{n+1} on p a proper p cannot be explicitly computed as in Proposition 4.2, but iteratively leading to the following p-adaptive algorithm.

pAPS-algorithm: Choose $\alpha_0 > 0$, $p := p_0 > 0$, and set $n := 0$.

1) Compute $u_{\alpha_n} \in \arg\min_{u \in X} \bar{E}_\tau(u, \alpha_n)$
2) Update $\alpha_{n+1} := \left(\frac{B_\tau(u_{\alpha_n})}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n$ if $H_\tau(u_{\alpha_n}) > 0$, and continue with step 3). Otherwise increase α_n, e.g., $\alpha_n := 10\alpha_n$, and go to step 1).
3) Compute $u_{\alpha_{n+1}} \in \arg\min_{u \in X} \bar{E}_\tau(u, \alpha_{n+1})$
4) a) if $H_\tau(u_{\alpha_0}) \leq B_\tau(u_{\alpha_0})$
   (i) if $H_\tau(u_{\alpha_{n+1}}) \leq B_\tau(u_{\alpha_{n+1}})$ go to step 5)
   (ii) if $H_\tau(u_{\alpha_{n+1}}) > B_\tau(u_{\alpha_{n+1}})$, decrease p, e.g., set $p := p/2$, and go to step 2)
   b) if $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$
   (i) if $H_\tau(u_{\alpha_{n+1}}) \geq B_\tau(u_{\alpha_{n+1}})$ go to step 5)
   (ii) if $H_\tau(u_{\alpha_{n+1}}) < B_\tau(u_{\alpha_{n+1}})$, decrease p, e.g., set $p := p/2$, and go to step 2)
5) Stop or set $n := n + 1$ and return to step 2).

The initial $p_0 > 0$ can be chosen arbitrarily. However, we suggest to choose it sufficiently large in order to keep the number of iterations small. In particular in our numerical experiments in Section 7 we set $p_0 = 32$, which seems large enough to us.

Proposition 4.6 *The pAPS-algorithm generates monotone sequences $(\alpha_n)_n$ and $(H_\tau(u_{\alpha_n}))_n$ such that $(H_\tau(u_{\alpha_n}))_n$ is bounded. Moreover, if $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$ or $B_\tau(u_\alpha) \leq \frac{1}{\tau}\|g - \langle g\rangle\|_\tau^\tau$ for all $\alpha > 0$, then $(\alpha_n)_n$ is also bounded.*

Proof If $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$, then by induction and Lemma 2.1 one shows that $0 < \alpha_{n+1} \leq \alpha_n$ and $0 \leq H_\tau(u_{\alpha_{n+1}}) \leq H_\tau(u_{\alpha_n})$ for all $n \in \mathbb{N}$. Consequently $(\alpha_n)_n$ and $(H_\tau(u_{\alpha_n}))_n$ are monotonically decreasing and bounded.

If $H_\tau(u_{\alpha_0}) \leq B_\tau(u_{\alpha_0})$, due to Lemma 2.1 we have that $0 < \alpha_n \leq \alpha_{n+1}$ and $H_\tau(u_{\alpha_n}) \leq H_\tau(u_{\alpha_{n+1}})$ for all $n \in \mathbb{N}$ and hence $(\alpha_n)_n$ and $(H_\tau(u_{\alpha_n}))_n$ are monotonically increasing. Since there exists $B_\tau^* > 0$ such that $H_\tau(u_{\alpha_n}) \leq B_\tau(u_{\alpha_n}) \leq B_\tau^*$ for all $n \in \mathbb{N}$, see Section 4.1, $(H_\tau(u_{\alpha_n}))_n$ is also bounded. If we additionally assume that $B_\tau(u_\alpha) \leq \frac{1}{\tau}\|g - \langle g\rangle\|_\tau^\tau$ for all $\alpha > 0$ and we set $B_\tau^* := \max_\alpha B_\tau(u_\alpha)$, then Theorem 2.2 ensures the existence of an $\alpha^* \geq 0$ such that $H_\tau(u_{\alpha^*}) = B_\tau^*$. By Lemma 2.1 it follows that $\alpha_n \leq \alpha^*$ for all $n \in \mathbb{N}$, since $H_\tau(u_{\alpha_n}) \leq B_\tau(u_{\alpha_n}) \leq B_\tau^*$. Hence, $(\alpha_n)_n$ is bounded, which finishes the proof. □

Since any monotone and bounded sequence converges to a finite limit, also $(\alpha_n)_n$ converges to a finite value if one of the assumptions in Proposition 4.6 holds. For constant B_τ we are even able to argue the convergence of the pAPS-algorithm to a solution of the constrained minimization problem (4.3).

Theorem 4.2 *Assume that $B_\tau(u) \equiv B_\tau$ is a constant independent of u and $\|g - \langle g\rangle\|_\tau^\tau \geq \nu_\tau |\Omega|$. Then the*

pAPS-algorithm generates a sequence $(\alpha_n)_n$ *such that* $\lim_{n\to\infty}\alpha_n = \bar{\alpha} > 0$ *with* $H_\tau(u_{\bar{\alpha}}) = \lim_{n\to\infty} H_\tau(u_{\alpha_n}) = B_\tau$ *and* $u_{\alpha_n} \to u_{\bar{\alpha}} \in \arg\min_{u\in X} \bar{E}_\tau(u,\bar{\alpha})$ *for* $n\to\infty$.

Proof We only consider the case when $H_\tau(u_{\alpha_0}) \le B_\tau$ by noting that the case $H_\tau(u_{\alpha_0}) > B_\tau$ can be shown analogous. By induction, we can show that $\alpha_n \le \alpha_{n+1}$ and $H_\tau(u_{\alpha_n}) \le H_\tau(u_{\alpha_{n+1}}) \le B_\tau$. More precisely, if $H_\tau(u_{\alpha_n}) \le B_\tau$ then $\alpha_{n+1} = \left(\frac{B_\tau}{H_\tau(u_{\alpha_n})}\right)^p \alpha_n \ge \alpha_n$, where $p > 0$ such that $H_\tau(u_{\alpha_{n+1}}) \le B_\tau$; cf. pAPS-algorithm. Then by Lemma 2.1 it follows that $H_\tau(u_{\alpha_n}) \le H_\tau(u_{\alpha_{n+1}}) \le B_\tau$. The rest of the proof is analog to the one of Theorem 4.1. □

Remark 4.1 The adaptive choice of the value p in the pAPS-algorithm is fundamental for proving convergence in Theorem 4.2. In particular, the value p is chosen in dependency of α, i.e., actually $p = p(\alpha)$, such that in the case of a constant B_τ the function $\alpha \mapsto \frac{H_\tau(u_\alpha)^{p(\alpha)}}{\alpha}$ is non-increasing; cf. Figure 7.2(a).

5 Locally constrained TV problem

In order to enhance image details, while preserving homogeneous regions, we formulate, as in [31,50], a locally constrained optimization problem. That is, instead of considering (4.2) we formulate

$$\min_{u\in X} R(u) \quad \text{s.t.} \quad S^\tau_{i,j}(u) \le \frac{\nu_\tau}{\tau} \ \text{ for all } x_{i,j} \in \Omega. \quad (5.1)$$

Here ν_τ is a fixed constant and

$$S^\tau_{i,j}(u) := \frac{1}{M_{i,j}} \sum_{x_{s,t}\in\mathcal{I}_{i,j}} \frac{1}{\tau}|(Tu)(x_{s,t}) - g(x_{s,t})|^\tau$$

denotes the local residual at $x_{i,j} \in \Omega$ with $\mathcal{I}_{i,j}$ being some suitable set of pixels around $x_{i,j}$ of size $M_{i,j}$, i.e., $M_{i,j} = |\mathcal{I}_{i,j}|$. For example, in [31,49,50] for $\mathcal{I}_{i,j}$ the set

$$\Omega^\omega_{i,j} = \left\{ x_{s+i,t+j} \in \Omega : -\frac{\omega-1}{2} \le s,t \le \frac{\omega-1}{2} \right\}$$

with a symmetric extension at the boundary and with ω being odd is used. That is, $\Omega^\omega_{i,j}$ is a set of pixels in a ω-by-ω window centered at $x_{i,j}$, i.e., $M_{i,j} = \omega^2$ for all i,j, such that $\Omega^\omega_{i,j} \not\subset \Omega$ for $x_{i,j}$ sufficiently close to $\partial\Omega$. Additionally we denote by $\tilde{\Omega}^\omega_{i,j}$ a set of pixels in a window centered at $x_{i,j}$ without any extension at the boundary, i.e.,

$$\tilde{\Omega}^\omega_{i,j} = \left\{ x_{s+i,t+j} : \max\left\{1-(i,j), -\frac{\omega-1}{2}\right\} \le (s,t) \le \min\left\{\frac{\omega-1}{2}, (N_1-i, N_2-j)\right\} \right\}.$$

Hence $\tilde{\Omega}^\omega_{i,j} \subset \Omega$ for all $x_{i,j} \in \Omega$. Before we analyze the difference between $\Omega^\omega_{i,j}$ and $\tilde{\Omega}^\omega_{i,j}$ with respect to the constrained minimization problem (5.1), we note that, since T does not annihilate constant functions, the existence of a solution of (5.1) is guaranteed; see [31, Theorem 2][50, Theorem 2].

Proposition 5.1 *(i) If* u *is a solution of* (5.1) *with* $\mathcal{I}_{i,j} = \Omega^\omega_{i,j}$, *then* $H_\tau(u) < B_\tau$.
(ii) If u *is a solution of* (5.1) *with* $\mathcal{I}_{i,j} = \tilde{\Omega}^\omega_{i,j}$, *then* $H_\tau(u) \le B_\tau$.

Proof (i) Since u is a solution of (5.1) and $\Omega^\omega_{i,j}$ is a set of pixels in a ω-by-ω window, we have

$$\begin{aligned} B_\tau &\ge \sum_{i,j} S^\tau_{i,j}(u) \\ &= \sum_{i,j} \frac{1}{\tau\omega^2} \sum_{x_{s,t}\in\Omega^\omega_{i,j}} |g(x_{s,t}) - Tu(x_{s,t})|^\tau \\ &> \frac{1}{\tau}\sum_{i,j} |g(x_{i,j}) - Tu(x_{i,j})|^\tau = H_\tau(u). \end{aligned}$$

Here we used that due to the sum over i,j each element (pixel) in $\Omega^\omega_{i,j}$ appears at most ω^2 times. More precisely, any pixel-coordinate in the set $\Lambda^\omega := \{(i,j) : \min\{i-1, j-1, N_1-i, N_2-j\} \ge \frac{\omega-1}{2}\}$ occurs exactly ω^2-times, while any other pixel-coordinate appears strictly less than ω^2-times. This shows the first statement.

(ii) For a minimizer u of (5.1) we obtain

$$\begin{aligned} B_\tau &\ge \sum_{i,j} S^\tau_{i,j}(u) \\ &= \sum_{i,j} \frac{1}{\tau M_{i,j}} \sum_{x_{s,t}\in\Omega^\omega_{i,j}} |g(x_{s,t}) - Tu(x_{s,t})|^\tau \\ &= \frac{1}{\tau}\sum_{i,j} |g(x_{i,j}) - Tu(x_{i,j})|^\tau = H_\tau(u), \end{aligned}$$

which concludes the proof. □

Note, that if $\mathcal{I}_{i,j} = \tilde{\Omega}^\omega_{i,j}$ then by Proposition 5.1 a minimizer of (5.1) also satisfies the constraint of the problem in (4.2) but is in general of course not a solution of (4.2).

Proposition 5.2 *Let* $\mathcal{I}_{i,j} = \tilde{\Omega}^\omega_{i,j}$, u_s *be a minimizer of* (4.2) *and* u_l *be a minimizer of* (5.1), *then* $R(u_s) \le R(u_l)$.

Proof Assume that $R(u_s) > R(u_l)$. Since u_s is a solution of (4.2) it satisfies the constraint $H_\tau(u_s) \le B_\tau$. By Proposition 5.1 we also have $H_\tau(u_l) \le B_\tau$. Since $R(u_s) > R(u_l)$, u_s is not the solution of (4.2) which is a contradiction. Hence, $R(u_s) \le R(u_l)$. □

Remark 5.1 Proposition 5.1 and its consequence are not special properties of the discrete setting. In a continuous setting, the locally constrained minimization problem is written (with a slight abuse of notations) as

$$\min_{u\in BV(\Omega)} \mathcal{R}(u) \;\text{ s.t. }\; \int_\Omega w(x,y)|Tu-g|^\tau(y)dy \le \nu_\tau \tag{5.2}$$

for almost every $x\in\Omega$, where now $T\in\mathcal{L}(L^2(\Omega))$, $g\in L^2(\Omega)$, and w is a normalized filter, i.e., $w\in L^\infty(\Omega\times\Omega)$, and $w\ge 0$ on $\Omega\times\Omega$ with

$$\begin{aligned}&\int_\Omega\int_\Omega w(x,y)dydx = 1\\ &\text{and}\\ &\int_\Omega\int_\Omega w(x,y)|\phi(y)|^\tau dydx \ge \epsilon\|\phi\|^\tau_{L^\tau(\Omega)}\end{aligned} \tag{5.3}$$

for all $\phi\in L^\tau(\Omega)$ and for some $\epsilon>0$ independent of ϕ; cf. [31,50]. If w is a filter such that the inequality in (5.3) becomes an equality with $\epsilon = 1/|\Omega|$, as it is the case in Proposition 5.1(ii), then a solution $\tilde{u}_l$ of the locally constrained minimization problem (5.2) satisfies

$$\mathcal{H}_\tau(\tilde{u}_l;g) \le \frac{\nu_\tau}{\tau}|\Omega| \;\text{ and }\; \mathcal{R}(\tilde{u}_l)\ge\mathcal{R}(\tilde{u}_s)$$

where $\tilde{u}_s$ is a solution of (2.1).

From Proposition 5.2 and Remark 5.1 we conclude, since $R(u_s)\le R(u_l)$ and $\mathcal{R}(\tilde{u}_s)\le\mathcal{R}(\tilde{u}_l)$, that u_s and $\tilde{u}_s$ are smoother than u_l and $\tilde{u}_l$, respectively. Hence the solution of the locally constrained minimization problem is expected to preserve details better than the minimizer of the globally constrained optimization problem.

5.1 Locally adaptive total variation algorithm

Whenever ν_τ depends on u problem (5.1) results in a quite nonlinear problem. Instead of considering nonlinear constraints we choose as in Section 4 a reference image $\tilde{u}$ and compute an approximate $\nu_\tau=\nu_\tau(\tilde{u})$. Then we are seeking for a solution u such that $S^\tau_{i,j}(u)$ is close to $\frac{\nu_\tau}{\tau}$.

We note, that for large $\alpha>0$ the minimization of (4.1) yields an over-smoothed restoration u_α and the residual contains details, i.e., we expect $H_\tau(u_\alpha)>B_\tau$. Hence, if $S^\tau_{i,j}(u_\alpha)>\frac{\nu_\tau}{\tau}$ we suppose that this is due to image details contained in the local residual image. In this situation we intend to decrease α in the local regions $\mathcal{I}_{i,j}$. In particular, we define, similar as in [31, 50], the local quantity $f^\omega_{i,j}$ by

$$f^\omega_{i,j} := \begin{cases} S^\tau_{i,j}(u_\alpha) & \text{if } S^\tau_{i,j}(u_\alpha)>\frac{\nu_\tau}{\tau},\\ \frac{\nu_\tau}{\tau} & \text{otherwise.}\end{cases}$$

Note, that $\frac{\nu_\tau}{\tau f^\omega_{i,j}}\le 1$ for all i,j and hence we set

$$\alpha(x_{i,j}) := \frac{1}{M_{i,j}}\sum_{x_{s,t}\in\mathcal{I}_{i,j}}\left(\frac{\nu_\tau}{\tau f^\omega_{s,t}}\right)^p\alpha(x_{s,t}). \tag{5.4}$$

On the other hand, for small $\alpha>0$ we get an under-smoothed image u_α, which still contains noise, i.e., we expect $H_\tau(u_\alpha)<B_\tau$. Analogously, if $S^\tau_{i,j}(u_\alpha)\le\frac{\nu_\tau}{\tau}$, we suppose that there is still noise left outside the residual image in $\mathcal{I}_{i,j}$. Hence we intend to increase α in the local regions $\mathcal{I}_{i,j}$ by defining

$$f^\omega_{i,j} := \begin{cases} S^\tau_{i,j}(u_\alpha) & \text{if } S^\tau_{i,j}(u_\alpha)<\frac{\nu_\tau}{\tau},\\ \frac{\nu_\tau}{\tau} & \text{otherwise,}\end{cases}$$

and setting α as in (5.4). Notice, that now $\frac{\nu_\tau}{\tau f^\omega_{i,j}}\ge 1$. These considerations lead to the following locally adapted total variation algorithm.

LATV-algorithm: Choose $\alpha_0>0$, $p:=p_0>0$, and set $n:=0$.

1) Compute $u_{\alpha_n}\in\arg\min_{u\in X}E_\tau(u,\alpha_n)$
2) (a) If $H_\tau(u_{\alpha_0})>B_\tau(u_{\alpha_0})$, then set

$$f^\omega_{i,j} := \max\left\{S^\tau_{i,j}(u_{\alpha_n}),\frac{\nu_\tau(u_{\alpha_n})}{\tau}\right\}$$

(b) If $H_\tau(u_{\alpha_0})\le B_\tau(u_{\alpha_0})$, then set

$$f^\omega_{i,j} := \min\left\{S^\tau_{i,j}(u_{\alpha_n}),\frac{\nu_\tau(u_{\alpha_n})}{\tau},\varepsilon\right\}$$

2) Update

$$\alpha_{n+1}(x_{i,j}) := \frac{1}{M_{i,j}}\sum_{x_{s,t}\in\mathcal{I}_{i,j}}\left(\frac{\nu_\tau(u_{\alpha_n})}{\tau f^\omega_{s,t}}\right)^p\alpha_n(x_{s,t}).$$

3) Stop or set $n:=n+1$ and return to step 1).

Here and below $\varepsilon>0$ is a small constant (e.g., in our experiments we choose $\varepsilon=10^{-14}$) to ensure that $f^\omega_{i,j}>0$, since it may happen that $S^\tau_{i,j}(u_{\alpha_n})=0$.

If $H_\tau(u_{\alpha_0})>B_\tau(u_{\alpha_0})$, we stop the algorithm as soon as the residual $H_\tau(u_{\alpha_n})<B_\tau(u_{\alpha_n})$ for the first time and set the desired locally varying $\alpha^*=\alpha_n$. If $H_\tau(u_{\alpha_0})\le B_\tau(u_{\alpha_0})$, we stop the algorithm as soon as the residual $H_\tau(u_{\alpha_n})>B_\tau(u_{\alpha_n})$ for the first time and set the desired locally varying $\alpha^*=\alpha_{n-1}$, since $H_\tau(u_{\alpha_{n-1}})\le\frac{\nu_\tau(u_{\alpha_{n-1}})}{\tau}$.

In contrast to the pAPS-algorithm here the power $p>0$ should be chosen sufficiently small, since p is not adapted during the iterations. Note, that small p only allow small changes of α in each iteration. In this way the algorithm is able the generate a function α^* such that $H_\tau(u_{\alpha^*})$ is very close to $\frac{\nu_\tau(u_{\alpha^*})}{\tau}$. On the contrary, small p have the drawback that the number of iterations till termination are kept large.

Proposition 5.3 *Assume $\mathcal{I}_{i,j} = \Omega^\omega_{i,j}$. If $\alpha_0 > 0$ such that $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$, then the LATV-algorithm generates a sequence $(\alpha_n)_n$ such that*

$$\sum_{i,j} \alpha_{n+1}(x_{i,j}) > \sum_{i,j} \alpha_n(x_{i,j}).$$

Proof By the same argument as in the proof of Proposition 5.1 we obtain

$$\begin{aligned}\sum_{i,j} \alpha_{n+1}(x_{i,j}) &= \sum_{i,j} \left(\frac{(\nu_\tau(u_{\alpha_n}))^p}{\tau^p \omega^2} \sum_{(s,t)\in\Omega^\omega_{i,j}} \frac{\alpha_n(x_{s,t})}{(f^\omega_{s,t})^p} \right) \\ &> \sum_{i,j} \left(\frac{(\nu_\tau(u_{\alpha_n}))^p \omega^2}{\tau^p \omega^2} \frac{\alpha_n(x_{i,j})}{(f^\omega_{i,j})^p} \right).\end{aligned}$$

Since $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$ we have by the LATV-algorithm that $f^\omega_{i,j} := \min\left\{ S^\tau_{i,j}(u_{\alpha_n}), \frac{\nu_\tau(u_{\alpha_n})}{\tau}, \varepsilon \right\} \le \frac{\nu_\tau(u_{\alpha_n})}{\tau}$ and hence we obtain $\sum_{i,j}(\alpha_{n+1})(x_{i,j}) > \sum_{i,j}(\alpha_n)(x_{i,j})$. □

Proposition 5.4 *Let $\mathcal{I}_{i,j} = \tilde{\Omega}^\omega_{i,j}$.*

(i) If $\alpha_0 > 0$ such that $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$, then the LATV-algorithm generates a sequence $(\alpha_n)_n$ such that

$$\sum_{i,j}(\alpha_{n+1})(x_{i,j}) \le \sum_{i,j}(\alpha_n)(x_{i,j}).$$

(ii) If $\alpha_0 > 0$ such that $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$, then the LATV-algorithm generates a sequence $(\alpha_n)_n$ such that

$$\sum_{i,j}(\alpha_{n+1})(x_{i,j}) \ge \sum_{i,j}(\alpha_n)(x_{i,j}).$$

Proof (i) By the same argument as in the proof of Proposition 5.1 and since $f^\omega_{i,j} := \max\left\{ S^\tau_{i,j}(u_{\alpha_n}), \frac{\nu_\tau(u_{\alpha_n})}{\tau} \right\} \ge \frac{\nu_\tau(u_{\alpha_n})}{\tau}$ we obtain

$$\begin{aligned}\sum_{i,j} \alpha_{n+1}(x_{i,j}) &= \sum_{i,j} \left(\frac{\nu_\tau(u_{\alpha_n})^p}{\tau^p M_{i,j}} \sum_{x_{s,t}\in\tilde{\Omega}^\omega_{i,j}} \frac{\alpha_n(x_{s,t})}{(f^\omega_{s,t})^p} \right) \\ &= \sum_{i,j} \left(\left(\frac{\nu_\tau(u_{\alpha_n})}{\tau} \right)^p \frac{\alpha_n(x_{i,j})}{(f^\omega_{i,j})^p} \right) \\ &\le \sum_{i,j} \alpha_n(x_{i,j}).\end{aligned}$$

(ii) Now $f^\omega_{i,j} := \min\left\{ S^\tau_{i,j}(u_{\alpha_n}), \frac{\nu_\tau(u_{\alpha_n})}{\tau}, \varepsilon \right\} \le \frac{\nu_\tau(u_{\alpha_n})}{\tau}$ and by the same arguments as above we get

$$\begin{aligned}\sum_{i,j} \alpha_{n+1}(x_{i,j}) &= \sum_{i,j} \left(\left(\frac{\nu_\tau(u_{\alpha_n})}{\tau} \right)^p \frac{\alpha_n(x_{i,j})}{(f^\omega_{i,j})^p} \right) \\ &\ge \sum_{i,j} \alpha_n(x_{i,j}).\end{aligned}$$

□

Similar as for the scalar case we make now the power p adaptively chosen.

pLATV-algorithm: Choose $\alpha_0 > 0$, $p := p_0 > 0$, and set $n := 0$.

0) Compute $u_{\alpha_n} \in \arg\min_{u\in X} E_\tau(u, \alpha_n)$
1) (a) If $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$, then set

$$f^\omega_{i,j} := \max\left\{ S^\tau_{i,j}(u_{\alpha_n}), \frac{\nu_\tau(u_{\alpha_n})}{\tau} \right\}$$

(b) If $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$, then set

$$f^\omega_{i,j} := \min\left\{ S^\tau_{i,j}(u_{\alpha_n}), \frac{\nu_\tau(u_{\alpha_n})}{\tau}, \varepsilon \right\}$$

2) Update

$$\alpha_{n+1}(x_{i,j}) := \frac{\alpha_n(x_{i,j})}{M_{i,j}} \sum_{x_{s,t}\in\mathcal{I}_{i,j}} \left(\frac{\nu_\tau(u_{\alpha_n})}{\tau f^\omega_{s,t}} \right)^p.$$

3) Compute $u_{\alpha_{n+1}} \in \arg\min_{u\in X} E_\tau(u, \alpha_n)$
4) (a) if $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$
(i) if $H_\tau(u_{\alpha_{n+1}}) \le B_\tau(u_{\alpha_{n+1}})$, go to step 5)
(ii) if $H_\tau(u_{\alpha_{n+1}}) > B_\tau(u_{\alpha_{n+1}})$, decrease p, e.g., set $p = p/10$, and go to step 2)
(b) if $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$
(i) if $H_\tau(u_{\alpha_{n+1}}) > B_\tau(u_{\alpha_{n+1}})$, go to step 5)
(ii) if $H_\tau(u_{\alpha_{n+1}}) \le B_\tau(u_{\alpha_{n+1}})$, decrease p, e.g., set $p = p/10$, and go to step 2)
5) Stop or set $n := n + 1$ and return to step 1).

In our numerical experiments this algorithm is terminated as soon as $|H_\tau(u_{\alpha_n}) - B_\tau(u_{\alpha_n})| \le 10^{-6}$ and $H_\tau(u_{\alpha_n}) \le B_\tau(u_{\alpha_n})$. Additionally we stop iterating when p is less than machine precision, since then anyway no progress is to expect. Due to the adaptive choice of p we obtain a monotonic behavior of the sequence $(\alpha_n)_n$.

Proposition 5.5 *The sequence $(\alpha_n)_n$ generated by the pLATV-algorithm is for any point $x \in \Omega$ monotone. In particular, it is monotonically decreasing for α_0 such that $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$, and monotonically increasing for α_0 such that $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$.*

Proof For $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$ we can show by induction that by the pLATV-algorithm $f^\omega_{i,j} \ge \frac{\nu_\tau(u_{\alpha_n})}{\tau}$ and hence $1 \ge \frac{\nu_\tau(u_{\alpha_n})}{\tau f^\omega_{i,j}}$ for all n. Then by the definition of α_{n+1} it follows

$$\begin{aligned}\alpha_{n+1}(x_{i,j}) &:= \frac{\alpha_n(x_{i,j})}{M_{i,j}} \sum_{x_{s,t}\in\mathcal{I}_{i,j}} \left(\frac{\nu_\tau(u_{\alpha_n})}{\tau f^\omega_{s,t}} \right)^p \\ &\le \alpha_n(x_{i,j}).\end{aligned}$$

By similar arguments we obtain for α_0 with $H_\tau(u_{\alpha_0}) \le B_\tau(u_{\alpha_0})$ that $\alpha_{n+1}(x_{i,j}) \ge \alpha_n(x_{i,j})$ for all $x_{i,j} \in \Omega$. □

In contrast to the SA-TV algorithm presented in [31,50], where the initial regularization parameter has to be chosen sufficiently small, in the LATV-algorithm as well as in the pLATV-algorithm the initial value α_0 can be chosen arbitrarily positive. However, in the case $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$ we cannot guarantee in general that the solution u_α obtained by the pLATV-algorithm fulfills $H_\tau(u_\alpha) \leq B_\tau(u_\alpha)$, not even if $B_\tau(\cdot)$ is constant, due to the stopping criterion with respect to the power p. On the contrary, if $H_\tau(u_{\alpha_0}) \leq B_\tau(u_{\alpha_0})$, then the pLATV-algorithm generates a sequence $(u_{\alpha_n})_n$ such that $H_\tau(u_{\alpha_n}) \leq B_\tau(u_{\alpha_n})$ for all n and hence also for the solution of the algorithm. As a consequence we would wish to choose $\alpha_0 > 0$ such that $H_\tau(u_{\alpha_0}) \leq B_\tau(u_{\alpha_0})$, which may be realized by the following simple automated procedure:

Algorithm 1: Input: $\alpha_0 > 0$ (arbitrary);

1) Compute $u_{\alpha_0} \in \arg\min_{u\in X} E_\tau(u, \alpha_0)$.
2) If $H_\tau(u_{\alpha_0}) > B_\tau(u_{\alpha_0})$ decrease α_0 by setting $\alpha_0 = c_{\alpha_0}\alpha_0$, with $c_{\alpha_0} \in (0,1)$, and continue with step 1), otherwise stop and return α_0.

6 Total Variation Minimization

In this section we are concerned with developing numerical methods for computing a minimizer of the discrete multi-scale L^τ-TV model, i.e.,

$$\min_{u\in X} E_\tau(u, \alpha). \tag{6.1}$$

6.1 L^2-TV minimization

Here we consider the case $\tau = 2$, i.e., the minimization problem

$$\min_{u\in X} \frac{1}{2}\|Tu - g\|_2^2 + R_\alpha(u), \tag{6.2}$$

and present solution methods, first for the case $T = I$ (image denoising) and then for general operators T including convolution type of operators.

6.1.1 An Algorithm for Image Denoising

If $T = I$, then (6.2) becomes an image denoising problem, i.e., the minimization problem

$$\min_{u\in X} \|u - g\|_2^2 + 2R_\alpha(u). \tag{6.3}$$

For solving this problem we use the algorithm of Chambolle and Pock [19], which leads to the following iterative scheme: Initialize $\tau, \sigma > 0$, $\theta \in [0,1]$, $(p_0, u_0) \in Y \times X$, set $\bar{u}_0 = u_0$, and iterate for $n \geq 0$

$$\mathbf{p}_{n+1}(x) = \frac{\mathbf{p}_n(x) + \sigma\nabla\bar{u}_n(x)}{\max\{\frac{1}{\alpha(x)}|\mathbf{p}_n(x) + \sigma\nabla\bar{u}_n(x)|, 1\}} \quad \forall x \in \Omega,$$
$$u_{n+1} = \frac{u_n + \tau \operatorname{div}\mathbf{p}_{n+1} + \tau g}{1+\tau},$$
$$\bar{u}_{n+1} = u_{n+1} + \theta(u_{n+1} - u_n),$$

where we choose $\theta = 1$ in our numerical experiments. In particular, in [19] it is shown that for $\theta = 1$ and $\tau\sigma\|\nabla\|^2 < 1$ the algorithm converges.

6.1.2 An algorithm for general operators T

Instead of minimizing (6.2) directly, we introduce the surrogate functional

$$\begin{aligned}\mathcal{S}(u,a) :&= \frac{1}{2}\|Tu - g\|_2^2 + R_\alpha(u) + \frac{\delta}{2}\|u - a\|_2^2 \\ &\quad - \frac{1}{2}\|T(u-a)\|_2^2 \\ &= \frac{\delta}{2}\|u - z(a)\|_2^2 + R_\alpha(u) + \psi(a, g, T),\end{aligned} \tag{6.4}$$

with $a, u \in X$, $z(a) = a - \frac{1}{\delta}T^*(Ta - g)$, ψ a function independent of u, and where we assume $\delta > \|T\|^2$. Note that

$$\min_{u\in X} \mathcal{S}(u,a) \Leftrightarrow \min_{u\in X} \|u - z(a)\|_2^2 + 2R_{\frac{\alpha}{\delta}}(u)$$

and hence to obtain a minimizer amounts to solve a minimization problem of the type (6.3) and can be solved as described in Section 6.1.1. Then an approximate solution of (6.2) can be computed by the following iterative algorithm:

$\mathbf{TV}_\alpha$-Algorithm: Choose $u_0 \in X$ and set $n := 0$.

1) Compute $u_{n+1} = \arg\min_{u\in X} \mathcal{S}(u, u_n)$
2) Stop or set $n := n + 1$ and return to step 1).

For scalar α it is shown in [24,27,28] that this iterative procedure generates a sequence $(u_n)_n$ which converges to a minimizer of (6.2). This convergence property can be easily extended to our non-scalar case yielding the following result.

Theorem 6.1 *For $\alpha : \Omega \to \mathbb{R}^+$ the TV_α-algorithm generates a sequence $(u_n)_n$ which converges to a solution of (6.2) for any initial choice of $u_0 \in X$.*

Proof A proof can be accomplished analogue to [28]. □

6.2 An algorithm for L^1-TV minimization

For computing a minimizer of the problem in (6.1) for $\tau = 1$, i.e.,

$$\min_{u\in X} \|Tu - g\|_1 + R_\alpha(u), \tag{6.5}$$

we generalize an algorithm that was proposed for L^1-TV minimization problems with a scalar regularization parameter $\alpha > 0$ in [5]. For this purpose we replace the original minimization (6.5) by

$$\min_{v,u\in X} \|v\|_1 + \frac{1}{2\gamma}\|Tu - g - v\|_2^2 + R_\alpha(u), \tag{6.6}$$

where $\gamma > 0$ is small, so that we have $g \approx Tu - v$. Actually, it can be shown that (6.6) converges to (6.5) as $\gamma \to 0$. In our experiments we actually choose $\gamma = 10^{-2}$. This leads to the following alternating algorithm.

L^1-TV$_\alpha$ algorithm: Initialize $\alpha > 0$, $u_0 \in X$ and set $n := 0$.

1) Compute

$$v_{n+1} = \arg\min_{v\in X} \|v\|_1 + \frac{1}{2\gamma}\|Tu_n - g - v\|_2^2$$

2) Compute

$$u_{n+1} \in \arg\min_{u\in X} \frac{1}{2\gamma}\|Tu - g - v_{n+1}\|_2^2 + R_\alpha(u)$$

3) Stop or set $n := n + 1$ and return to step 1).

The minimizer v_{n+1} in step 1) of the L^1-TV$_\alpha$ algorithm can be easily computed via a soft-thresholding, i.e., $v_{n+1} = \mathrm{ST}(Tu_n - g, \gamma)$, where

$$\mathrm{ST}(g,\gamma)(x) = \begin{cases} g(x) - \gamma & \text{if } g(x) > \gamma, \\ 0 & \text{if } |g(x)| \le \gamma, \\ g(x) + \gamma & \text{if } g(x) < -\gamma \end{cases}$$

for all $x \in \Omega$. The minimization problem in step 2) is equivalent to

$$\arg\min_{u\in X} \frac{1}{2}\|Tu - g - v_{n+1}\|_2^2 + R_{\gamma\alpha}(u) \tag{6.7}$$

and hence is of the type (6.2). Thus an approximate solution of (6.7) can be computed as described above; see Section 6.1.

Theorem 6.2 *The sequence $(u_n, v_n)_n$ generated by the L^1-TV$_\alpha$ algorithm converges to a minimizer of* (6.6).

Proof The statement can be shown analogue to [5]. □

6.3 A primal-dual method for L^1-TV minimization

For deriving a primal-dual algorithm we consider instead of

$$\min_{u\in X} \|Tu - g\|_1 + R_\alpha(u)$$

a regularized version similar as in [50], i.e.,

$$\min_{u\in X} \sum_{x\in\Omega} \frac{\kappa}{2}|\nabla u(x)|_{l^2}^2 + \Phi^*_{\mu,\beta}(-Tu(x)) + \Psi^*_\gamma(\nabla u(x)),$$

where

$$\Phi^*_{\mu,\beta}(v) := \begin{cases} \frac{1}{2\beta}|v(x) - g(x)|^2 & \text{if } |v(x) - g(x)| \le \beta, \\ \tilde\Phi^*_{\mu,\beta}(v) & \text{if } |v(x) - g(x)| > \beta, \end{cases}$$

with $\tilde\Phi^*_{\mu,\beta}(v) := \frac{1}{2(\beta+\mu)}|v(x)-g(x)|^2 + \frac{\mu}{\beta+\mu}|v(x)-g(x)| - \frac{\mu\beta}{2(\beta+\mu)}$ and

$$\Psi^*_\gamma(\mathbf{p}) := \begin{cases} \frac{1}{2\gamma}|\mathbf{p}(x)|_{l^2}^2 & \text{if } |\mathbf{p}(x)|_{l^2} < \alpha(x)\gamma, \\ \alpha(x)|\mathbf{p}(x)|_{l^2} - \frac{\gamma}{2}|\alpha(x)|^2 & \text{if } |\mathbf{p}(x)|_{l^2} \ge \alpha(x)\gamma, \end{cases}$$

see [50] for more details. Here κ, β, μ, and γ are fixed positive constants. The corresponding optimality conditions for the coupled solutions $(\bar u, \bar{\mathbf{p}})$ due to the Fenchel theorem [33] are given by

$$\begin{aligned} -\operatorname{div}\bar{\mathbf{p}}(x) &= -\kappa\Delta\bar u(x) + \frac{1}{\beta+\mu}T^*(T\bar u(x) - g(x)) \\ &\quad + \frac{\mu}{\beta+\mu}T^*\frac{T\bar u(x) - g(x)}{\max\{\beta, |T\bar u(x) - g(x)|\}} \\ -\bar{\mathbf{p}}(x) &= \frac{1}{\gamma}\nabla\bar u(x) \quad \text{if } |\bar{\mathbf{p}}(x)|_{l^2} < \alpha(x) \\ -\bar{\mathbf{p}}(x) &= \alpha(x)\frac{\nabla\bar u(x)}{|\nabla\bar u(x)|_{l^2}} \quad \text{if } |\bar{\mathbf{p}}(x)|_{l^2} = \alpha(x), \end{aligned}$$

for all $x \in \Omega$. The latter two conditions can be summarized to $-\bar{\mathbf{p}}(x) = \frac{\alpha(x)\nabla\bar u(x)}{\max\{\gamma\alpha(x), |\nabla\bar u(x)|_{l^2}\}}$. Then setting $\bar{\mathbf{q}} = -\bar{\mathbf{p}}$ and $\bar v = \frac{T\bar u - g}{\max\{\beta, |T\bar u - g|\}}$ leads to the following system of equation:

$$\begin{aligned} 0 &= -\max\{\beta, |T\bar u(x) - g(x)|\}\bar v + T\bar u(x) - g(x) \\ 0 &= \operatorname{div}\bar{\mathbf{q}}(x) + \kappa\Delta\bar u(x) - \frac{1}{\beta+\mu}T^*(T\bar u(x) - g(x)) \\ &\quad - \frac{\mu}{\beta+\mu}T^*\bar v(x) \\ 0 &= \max\{\gamma\alpha(x), |\nabla\bar u(x)|_{l^2}\}\bar{\mathbf{q}}(x) - \alpha(x)\nabla\bar u(x) \end{aligned} \tag{6.8}$$

for all $x \in \Omega$. This system can be solved efficiently by a semi-smooth Newton algorithm; see Appendix A for a description of the method and for the choice of the parameters κ, β, μ, and γ.

7 Numerical Experiments

In the following we present numerical experiments for studying the behavior of the proposed algorithms (i,e., APS-, pAPS-, LATV-, pLATV-algorithm) with respect to its image restoration capabilities and its stability concerning the choice of the initial value α_0. The performance of these methods is compared quantitatively by means of the peak signal-to-noise-ratio (PSNR) [11], which is widely used as an image quality assessment measure, and the structural similarity measure (MSSIM) [78], which relates to perceived visual quality better than PSNR. When an approximate solution of the L^1-TV model is calculated, we also compare the restorations by the mean absolute error (MAE), which is an L^1-based measure defined as

$$\text{MAE} = \frac{1}{\|u - \hat{u}\|_1},$$

where $\hat{u}$ denotes the true image and u represents the obtained restoration. In general, when comparing PSNR and MSSIM, large values indicate better reconstruction than smaller values, while the smaller MAE becomes the better the reconstruction results are.

Whenever an image is corrupted by Gaussian noise we compute a solution by means of the (multi-scale) L^2-TV model, while for images containing impulsive noise the (multi-scale) L^1-TV model is always considered.

In our numerical experiments the CPS-, APS-, and pAPS-algorithm are terminated as soon as

$$\frac{|H_\tau(u_{\alpha_n}) - B_\tau(u_{\alpha_n})|}{B_\tau(u_{\alpha_n})} \leq \epsilon_B = 10^{-5}$$

or the norm of the difference of two successive iterates $\alpha^{(n)}$ and $\alpha^{(n+1)}$ drops below the threshold $\epsilon_\alpha = 10^{-10}$, i.e., $\|\alpha^{(n)} - \alpha^{(n+1)}\| < \epsilon_\alpha$. The latter stopping criterion is used to terminate the algorithms if $(\alpha_n)_n$ stagnates and only very little progress is to expect. In fact, if our algorithm converges at least linearly, i.e., if there exists an $\varepsilon_\alpha \in (0,1)$ and an $m > 0$ such that for all $n \geq m$ we have $\|\alpha^{(n+1)} - \alpha^{(\infty)}\| < \varepsilon_\alpha \|\alpha^{(n)} - \alpha^{(\infty)}\|$, the second stopping criterion at least ensures that the distance between our obtained result α and $\alpha^{(\infty)}$ is $\|\alpha - \alpha^{(\infty)}\| \leq \frac{\epsilon_\alpha \varepsilon_\alpha}{1-\varepsilon_\alpha}$.

7.1 Automatic scalar parameter selection

For automatically selecting the scalar parameter α in (4.1) we presented in Section 4 the APS- and pAPS-algorithm. Here we compare their performance for image denoising and image deblurring.

7.1.1 Gaussian noise removal

For recovering images corrupted by Gaussian noise with mean zero and standard deviation σ we minimize the functional in (4.1) by setting $\tau = 2$ and $T = I$. Then $B_2 = \frac{\sigma^2}{2}|\Omega|$ is a constant independent of u. The automatic selection of a suitable regularization parameter α is here performed by the CPS-, APS-, and pAPS-algorithm, where the contained minimization problem is solved by the method presented in Section 6.1.1. We recall, that by [16, Theorem 4] and Theorem 4.2 it is ensured that the CPS- and the pAPS-algorithm generate sequences $(\alpha_n)_n$ which converge to $\bar{\alpha}$ such that $u_{\bar{\alpha}}$ solves (4.3). In particular, in the pAPS-algorithm the value p is chosen in dependency of α, i.e., $p = p(\alpha)$, such that $\alpha \to \frac{(H_\tau(u_\alpha))^{p(\alpha)}}{\alpha}$ is non-increasing, see Figure 7.2(a). This property is fundamental for obtaining convergence of this algorithm; see Theorem 4.2. For the APS-algorithm such a monotonic behavior is not guaranteed and hence we cannot ensure its convergence. Nevertheless, if the APS-algorithm generates α's such that the function $\alpha \to \frac{H_\tau(u_\alpha)}{\alpha}$ is non-increasing, then it indeed converges to the desired solution, see Theorem 4.1. Unfortunately, the non-increase of the function $\alpha \to \frac{H_\tau(u_\alpha)}{\alpha}$ does not hold always, see Figure 7.2(b).

For a performance-comparison here we consider the phantom-image of size 256×256 pixels, see Figure 7.1(a), corrupted only by Gaussian white noise with different noise-levels, i.e., $\sigma \in \{0.3, 0.1, 0.05, 0.01\}$. In the pAPS-algorithm we set $p_0 = 32$. The obtained results for different initial α_0, i.e., $\alpha_0 \in \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ are summarized in Table 7.1 - 7.4. We observe that all three methods converge to the same regularization parameter α and hence generate results with the same PSNR and MSSIM. Hence, in these experiments, despite the lack of theoretical convergence, also the APS-algorithm seems to converge to the desired solutions.

Looking at the number of iterations needed till termination, we observe from Table 7.1 - 7.4 that the APS-algorithm always needs significantly less iterations than the CPS-algorithm till termination. This is attributed to the different updates of α. Recall, that for a fixed α_n in the CPS-algorithm we set $\alpha_{n+1}^{CPS} := \sqrt{\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})}}\alpha_n$, while in the APS-algorithm we have $\alpha_{n+1}^{APS} := \frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})}\alpha_n$. Now, note that $\sqrt{\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})}} \leq \frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})}$, if $\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})} \geq 1$ and $\sqrt{\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})}} \geq \frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})}$, if $\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})} \leq 1$. Hence, we obtain $\alpha_n \leq \alpha_{n+1}^{CPS} \leq \alpha_{n+1}^{APS}$ if $\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})} \geq 1$ and $\alpha_n \geq \alpha_{n+1}^{CPS} \geq \alpha_{n+1}^{APS}$ if $\frac{\nu_2|\Omega|}{\tau H_2(u_{\alpha_n})} \leq 1$. That is, in the APS-algorithm α changes more significantly in each iteration than in the CPS-algorithm, which leads to a

Table 7.1 *Reconstruction of the phantom-image corrupted by Gaussian white noise with $\sigma = 0.3$. In the pAPS-algorithm we set $p_0 = 32$.*

	CPS				APS				pAPS			
α_0	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α
1	19.84	0.7989	55	0.2381	19.84	0.7989	25	0.2381	19.84	0.7989	8	0.2381
10^{-1}	19.84	0.7989	42	0.2381	19.84	0.7989	18	0.2381	19.84	0.7989	18	0.2381
10^{-2}	19.84	0.7989	43	0.2381	19.84	0.7989	27	0.2381	19.84	0.7989	43	0.2381
10^{-3}	19.84	0.7989	43	0.2381	19.84	0.7989	31	0.2381	19.84	0.7989	43	0.2381
10^{-4}	19.84	0.7989	43	0.2381	19.84	0.7989	35	0.2381	19.84	0.7989	43	0.2381

Table 7.2 *Reconstruction of the phantom-image corrupted by Gaussian white noise with $\sigma = 0.1$. In the pAPS-algorithm we set $p_0 = 32$.*

	CPS				APS				pAPS			
α_0	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α
1	28.97	0.9644	47	0.0786	28.97	0.9644	21	0.0786	28.97	0.9644	21	0.0786
10^{-1}	28.97	0.9644	38	0.0786	28.97	0.9644	17	0.0786	28.97	0.9644	6	0.0786
10^{-2}	28.97	0.9644	40	0.0786	28.97	0.9644	20	0.0786	28.97	0.9644	40	0.0786
10^{-3}	28.97	0.9644	41	0.0786	28.97	0.9644	22	0.0786	28.97	0.9644	41	0.0786
10^{-4}	28.97	0.9644	41	0.0786	28.97	0.9644	23	0.0786	28.97	0.9644	41	0.0786

Table 7.3 *Reconstruction of the phantom-image corrupted by Gaussian white noise with $\sigma = 0.05$. In the pAPS-algorithm we set $p_0 = 32$.*

	CPS				APS				pAPS			
α_0	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α
1	34.97	0.9887	46	0.0394	34.97	0.9887	20	0.0394	34.97	0.9887	20	0.0394
10^{-1}	34.97	0.9887	44	0.0394	34.97	0.9887	20	0.0394	34.97	0.9887	20	0.0394
10^{-2}	34.97	0.9887	40	0.0394	34.97	0.9887	18	0.0394	34.97	0.9887	40	0.0394
10^{-3}	34.97	0.9887	41	0.0394	34.97	0.9887	19	0.0394	34.97	0.9887	41	0.0394
10^{-4}	34.97	0.9887	41	0.0394	34.97	0.9887	22	0.0394	34.97	0.9887	41	0.0394

faster convergence with respect to the number of iterations. Nevertheless, exactly this behavior allows the function $\alpha \to \frac{H_2(u_\alpha)}{\alpha}$ to increase which is responsible that the convergence of the APS-algorithm is not guaranteed in general. However, in our experiments we observed that the function $\alpha \to \frac{H_2(u_\alpha)}{\alpha}$ only increases in the first iterations, but non-increases (actually even decreases) afterwards, see Figure 7.2(b). This is actually enough to guarantee convergence, as discussed in Section 4, since we can consider the solution of the last step in which the desired monotonic behavior is not fulfilled as a "new" initial value. Since from this point on the non-increase holds, we get convergence of the algorithm.

The pAPS-algorithm is designed to ensure the non-increase of the function $\alpha \to \frac{(H_\tau(u_\alpha))^{p(\alpha)}}{\alpha}$ by choosing $p(\alpha)$ in each iteration accordingly, which is done by the algorithm automatically. If $p(\alpha) = p = 1/2$ in each iteration than the pAPS-algorithm becomes the CPS-algorithm, as it happens sometimes in practice; see Table 7.1 - 7.4. Since the CPS-algorithm converges [16], the pAPS-algorithm always yields $p \geq 1/2$. In particular, we observe that if the starting value α_0 is larger than the requested regularization parameter α, less iteration till termination are needed than with the CPS-algorithm. On the contrary, if α_0 is smaller than the desired α, $p = 1/2$ is chosen by the algorithm to ensure the monotonicity. Similar behaviors as described above are also observed for denoising other and real images as well.

All three approaches converge to the same solution. However, the APS- and pAPS-algorithm improve the CPS-algorithm in the following way for image denoising: They need less or at least the same number of iterations till convergence and, while the CPS-algorithm is restricted to denoising images corrupted by Gaussian white noise, i.e., to the L^2-TV model with $T = I$, the APS- and pAPS-algorithm are much more general and can handle convolution type of problems as well as removing impulsive noise via an L^1-data term.

Table 7.4 *Reconstruction of the phantom-image corrupted by Gaussian white noise with $\sigma = 0.01$. In the pAPS-algorithm we set $p_0 = 32$.*

	CPS				APS				pAPS			
α_0	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α
1	48.88	0.9994	47	0.0079	48.88	0.9994	17	0.0079	48.88	0.9994	47	0.0079
10^{-1}	48.88	0.9994	46	0.0079	48.88	0.9994	18	0.0079	48.88	0.9994	46	0.0079
10^{-2}	48.88	0.9994	39	0.0079	48.88	0.9994	17	0.0079	48.88	0.9994	6	0.0079
10^{-3}	48.88	0.9994	41	0.0079	48.88	0.9994	20	0.0079	48.88	0.9994	41	0.0079
10^{-4}	48.88	0.9994	41	0.0079	48.88	0.9994	19	0.0079	48.88	0.9994	41	0.0079

7.1.2 Image deblurring

Now, we consider the situation when an image is corrupted by some additive Gaussian noise and additionally blurred. Then the operator T is chosen according to the blurring kernel, which we assume here to be known. For testing the APS- and pAPS-algorithm in this case we take the cameraman-image of Figure 7.1(b), which is of size 256×256 pixels, blur it by a Gaussian blurring kernel of size 5×5 pixels and standard deviation 10 and additionally add some Gaussian white noise with variance σ^2. The minimization problem in the APS- and pAPS-algorithm is solved approximately by the TV_α-algorithm. In Table 7.5 the PSNR and MSSIM of the reconstructions for different σ's, i.e., $\sigma \in \{0.3, 0.1, 0.05\}$, and different α_0's, i.e., $\alpha_0 \in \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ are presented. Moreover, the table also contains the number of iterations needed until termination and the obtained regularization parameter α of the APS- and pAPS-algorithm. In these tests both algorithms seem to converge to the same minimizer and regularization parameter. However, the pAPS-algorithm needs much less iterations than the APS-algorithm till termination. This behavior might be attributed to the choice of the power p in the pAPS-algorithm, since we observe in all our experiments that $p > 1$ till termination.

7.1.3 Impulsive noise removal

It has been demonstrated that for removing impulsive noise in images one should minimize the L^1-TV model rather than the L^2-TV model. Then for calculating a suitable regularization parameter α in the L^1-TV model we use the APS- and pAPS-algorithm, in which the minimization problems are solved approximately by the L^1-TV_α-algorithm. Here, we consider the cameraman-image corrupted by salt-and-pepper noise or random-valued impulse noise with different noise-levels, i.e., $r_1 = r_2 \in \{0.3, 0.1, 0.05\}$ and $r \in \{0.3, 0.1, 0.05\}$ respectively. The obtained results for different α_0's are summarized in Table 7.6 and Table 7.7. For the removal of salt-and-pepper noise we observe from Table 7.6 similar behaviors of the APS- and pAPS-algorithm as above for removing Gaussian noise. In particular, both algorithms converge to the same solutions. However, in many cases the APS-algorithm needs significantly less iterations than the pAPS-algorithm. These behaviors are also observed in Table 7.7 for removing random-valued impulse noise as long as the APS-algorithm finds a solution. In fact, for $r = 0.05$ it actually does not converge but oscillates as depicted in Figure 7.3. Blank spaces in Table 7.7 mean, that the algorithm did not converge for this setting.

7.2 Locally adaptive total variation minimization

In this section various experiments are presented to evaluate the performance of the LATV- and pLATV-algorithm presented in Section 5. Their performance is compared with the proposed pAPS-algorithm as well as with the SA-TV-algorithm introduced in [31] for L^2-TV minimization and in [50] for L^1-TV minimization. We recall that the SA-TV methods perform an approximate solution for the optimization problem in (1.10), respectively, and compute automatically a spatially varying λ based on a local variance estimation. However, as pointed out in [31,50], they only perform efficiently when the initial λ is chosen sufficiently small, as we will do in our numerics. On the contrary, for the LATV- and pLATV-algorithm any positive initial α_0 is sufficient.

For the comparison we consider four different images, shown in Figure 7.1, which are all of size 256×256 pixels. In all our experiments for the SA-TV-algorithm we use $\mathcal{I}_{i,j} = \Omega^\omega_{i,j}$, see [31], and we set the window-size to 11×11 pixels. For the LATV- and pLATV-algorithm we use the same window-size, i.e., $\omega = 11$, if not otherwise specified, and choose $p_0 = \frac{1}{2}$.

7.3 Gaussian noise removal

We start this section by investigating the stability of the SA-TV-, LATV-, and pLATV-algorithm with respect to the initial regularization parameter, i.e., λ_0

Table 7.5 *Reconstruction of the cameraman-image corrupted by Gaussian blurring kernel of size* 5×5 *and standard deviation* 10. *In the pAPS-algorithm we set* $p_0 = 32$.

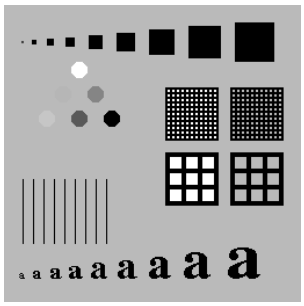
		APS				pAPS			
α_0	σ	PSNR	MSSIM	It	α	PSNR	MSSIM	It	α
1	0.3	21.04	0.6410	261	0.2219	21.04	0.6410	13	0.2219
10^{-1}	0.3	21.04	0.6410	244	0.2218	21.04	0.6410	12	0.2218
10^{-2}	0.3	21.04	0.6410	271	0.2218	21.04	0.6410	29	0.2218
10^{-3}	0.3	21.04	0.6410	275	0.2218	21.04	0.6410	65	0.2218
10^{-4}	0.3	21.04	0.6410	278	0.2218	21.04	0.6410	29	0.2218
1	0.1	23.11	0.7174	149	0.0514	23.11	0.7174	33	0.0514
10^{-1}	0.1	23.11	0.7174	135	0.0514	23.11	0.7174	14	0.0514
10^{-2}	0.1	23.11	0.7175	151	0.0514	23.11	0.7174	4	0.0514
10^{-3}	0.1	23.11	0.7175	161	0.0514	23.11	0.7175	37	0.0514
10^{-4}	0.1	23.11	0.7175	164	0.0514	23.11	0.7175	12	0.0514
1	0.05	24.14	0.7562	106	0.0183	24.14	0.7562	50	0.0183
10^{-1}	0.05	24.14	0.7562	104	0.0183	24.14	0.7562	23	0.0183
10^{-2}	0.05	24.14	0.7562	106	0.0183	24.14	0.7562	10	0.0183
10^{-3}	0.05	24.14	0.7562	125	0.0183	24.14	0.7562	11	0.0183
10^{-4}	0.05	24.13	0.7562	130	0.0183	24.14	0.7562	11	0.0183

Table 7.6 *Reconstruction of the cameraman-image corrupted by salt-and-pepper noise* $r_1 = r_2$. *In the pAPS-algorithm we set* $p_0 = 32$.

		APS					pAPS				
α_0	$r_1 = r_2$	PSNR	MSSIM	It	α	MAE	PSNR	MSSIM	It	α	MAE
1	0.3	20.72	0.6483	73	0.8147	0.0415	20.72	0.6483	13	0.8147	0.0415
10^{-1}	0.3	20.72	0.6483	110	0.8147	0.0415	20.72	0.6483	250	0.8146	0.0415
10^{-2}	0.3	20.72	0.6483	109	0.8147	0.0415	20.72	0.6483	118	0.8146	0.0415
10^{-3}	0.3	20.72	0.6483	109	0.8147	0.0415	20.72	0.6483	250	0.8146	0.0415
10^{-4}	0.3	20.72	0.6483	109	0.8147	0.0415	20.72	0.6483	508	0.8146	0.0415
1	0.1	23.35	0.7292	11	0.5076	0.0233	23.35	0.7292	9	0.5076	0.0233
10^{-1}	0.1	23.35	0.7292	18	0.5076	0.0233	23.35	0.7292	75	0.5076	0.0233
10^{-2}	0.1	23.35	0.7292	18	0.5076	0.0233	23.35	0.7292	26	0.5076	0.0233
10^{-3}	0.1	23.35	0.7292	18	0.5076	0.0233	23.35	0.7292	34	0.5076	0.0233
10^{-4}	0.1	23.35	0.7292	18	0.5076	0.0233	23.35	0.7292	84	0.5076	0.0233
1	0.05	24.97	0.7968	11	0.4101	0.0158	24.97	0.7968	9	0.4101	0.0158
10^{-1}	0.05	24.97	0.7968	14	0.4101	0.0158	24.97	0.7968	94	0.4101	0.0158
10^{-2}	0.05	24.97	0.7968	14	0.4101	0.0158	24.97	0.7968	45	0.4101	0.0158
10^{-3}	0.05	24.97	0.7968	14	0.4101	0.0158	24.97	0.7968	20	0.4101	0.0158
10^{-4}	0.05	24.97	0.7968	14	0.4101	0.0158	24.97	0.7968	94	0.4101	0.0158

Table 7.7 *Reconstruction of the cameraman-image corrupted by random-valued impulse noise. In the pAPS-algorithm we set* $p_0 = 32$.

		APS					pAPS				
α_0	r	PSNR	MSSIM	It	α	MAE	PSNR	MSSIM	It	α	MAE
1	0.3	23.17	0.6959	12	0.5152	0.0298	23.17	0.6959	12	0.5152	0.0298
10^{-1}	0.3	23.17	0.6959	18	0.5152	0.0298	23.17	0.6959	66	0.5152	0.0298
10^{-2}	0.3	23.17	0.6959	18	0.5152	0.0298	23.17	0.6959	29	0.5152	0.0298
10^{-3}	0.3	23.17	0.6959	18	0.5152	0.0298	23.17	0.6959	31	0.5152	0.0298
10^{-4}	0.3	23.17	0.6959	18	0.5152	0.0298	23.17	0.6959	73	0.5152	0.0298
1	0.1	26.28	0.7868	8	0.3886	0.0151	26.28	0.7868	10	0.3886	0.0151
10^{-1}	0.1	26.28	0.7868	10	0.3886	0.0151	26.28	0.7868	30	0.3886	0.0151
10^{-2}	0.1	26.28	0.7868	10	0.3886	0.0151	26.28	0.7868	67	0.3886	0.0151
10^{-3}	0.1	26.28	0.7868	10	0.3886	0.0151	26.28	0.7868	66	0.3886	0.0151
10^{-4}	0.1	26.28	0.7868	10	0.3886	0.0151	26.28	0.7868	70	0.3886	0.0151
1	0.05						26.60	0.7546	9	0.3492	0.0127
10^{-1}	0.05						26.60	0.7546	94	0.3492	0.0127
10^{-2}	0.05						26.60	0.7546	45	0.3492	0.0127
10^{-3}	0.05						26.60	0.7546	20	0.3492	0.0127
10^{-4}	0.05						26.60	0.7546	94	0.3492	0.0127



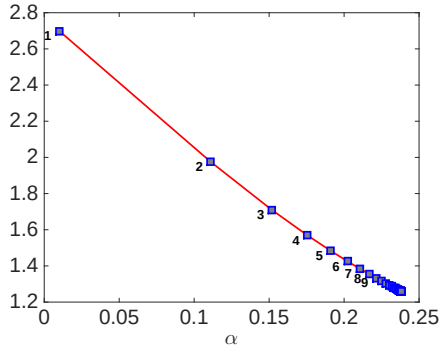
(a) phantom

(b) cameraman

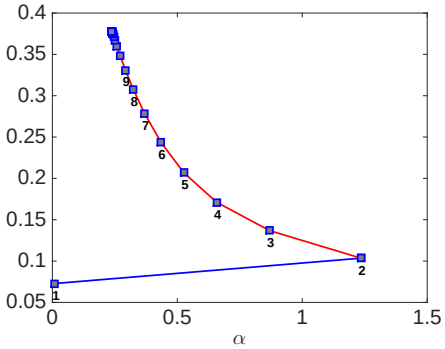
(c) barbara

(d) lena

Fig. 7.1 *Original images.*



(a)



(b)

Fig. 7.2 *Denoising of the phantom-image corrupted with Gaussian white noise with $\sigma = 0.03$. (a) Plot of the function $\alpha \to \frac{H_\tau(u_\alpha)^p}{\alpha}$ of the pAPS-algorithm with $\alpha_0 = 10^{-2}$. (b) Plot of the function $\alpha \to \frac{H_\tau(u_\alpha)}{\alpha}$ of the APS-algorithm with $\alpha_0 = 10^{-2}$. The desired monotone behavior is observed after the second iteration (red part of the curve).*

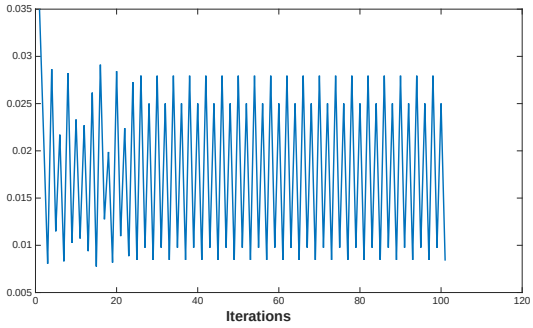


Fig. 7.3 *Progress of $H_\tau(u_{\alpha_n})$ of the APS-algorithm for removing random-valued impulse noise with $r = 0.05$.*

for the SA-TV-algorithm and α_0 for the other algorithms, by denoising the cameraman-image corrupted by Gaussian white noise with standard deviation $\sigma = 0.1$. In this context we also compare the difference of the pLATV-algorithm with and without using Algorithm 1 for computing automatically an initial param-

eter, where we set $c_{\alpha_0} = \frac{1}{5}$. The minimization problems contained in the LATV- and pLATV-algorithm are solved as described in Section 6.1.1. For comparison reasons we define the values $\mathrm{PSNR}_{\mathrm{diff}} := \max_{\alpha_0} \mathrm{PSNR}(\alpha_0) - \min_{\alpha_0} \mathrm{PSNR}(\alpha_0)$ and $\mathrm{MSSIM}_{\mathrm{diff}} := \max_{\alpha_0} \mathrm{MSSIM}(\alpha_0) - \min_{\alpha_0} \mathrm{MSSIM}(\alpha_0)$ to measure the variation of the considered quality measures. Here $\mathrm{PSNR}(\alpha_0)$ and $\mathrm{MSSIM}(\alpha_0)$ are the PSNR and MSSIM values of the reconstructions, which are obtained from the considered algorithms when the initial regularization parameter is set to α_0. From Table 7.8 we observe that the pLATV-algorithm with and without Algorithm 1 are more stable with respect to the initial regularization parameter than the LATV-algorithm and the SA-TV-algorithm. This stable performance of the pLATV-algorithm is reasoned by the adaptivity of the value p, which allows the algorithm to reach the desired residual (at least very closely) for any α_0. As expected, the pLATV-algorithm with Algorithm 1 is even more stable with respect to α_0 than the pLATV-algorithm alone, since due to Algorithm 1 the difference of the actually used initial parameters in the pLATV-algorithm is rather small leading to very similar results. Note, that if α_0 is sufficiently small, then the pLATV-algorithm with and without Algorithm 1 coincide, see Table 7.8 for $\alpha_0 \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. Actually in the rest of our experiments we choose α_0 always so small such that Algorithm 1 returns the inputted α_0.

In Table 7.9 we report on the performance-tests of the pLATV-algorithm with respect to the chosen type of window, i.e., $\mathcal{I}_{i,j} = \tilde{\Omega}^{\omega}_{i,j}$ and $\mathcal{I}_{i,j} = \Omega^{\omega}_{i,j}$. We observe that independently which type of window is used the algorithm finds nearly the same reconstruction. This may be attributed to the fact that the windows in the interior are the same for both types of window. Nevertheless, the boundaries are treated differently, which leads to different theoretical results, but seems not to have significant influence on the practical behavior. A similar behavior is observed for the LATV-algorithm, as the LATV- and pLATV-algorithm return nearly the same reconstructions as observed below in Table 7.10. Since for both types of windows nearly the same results are obtained, in the rest of our experiments we limit ourselves to always set $\mathcal{I}_{i,j} = \tilde{\Omega}^{\omega}_{i,j}$ in the LATV- and pLATV-algorithm.

Now we test the algorithms for different images corrupted by Gaussian noise with zero mean and different standard deviations σ, i.e., $\sigma \in \{0.3, 0.1, 0.05, 0.01\}$. The initial regularization parameter α_0 is set to 10^{-4} in the pAPS-, LATV-, and pLATV-algorithm. In the SA-TV-algorithm we choose $\lambda_0 = 10^{-4}$, which seems sufficiently small. From Table 7.10 we observe that all considered algorithms behave very similar. However, for $\sigma \in \{0.1, 0.05, 0.01\}$ the SA-TV-algorithm most of the times performs best with respect to PSNR and MSSIM, while sometimes the LATV- and pLATV-algorithm have larger PSNR and MSSIM. That is, looking at these quality measures a locally varying regularization weight is preferred to a scalar one, as long as σ is sufficiently small. In Figure 7.4 we present the reconstructions obtained via the considered algorithms and we observe that the LATV- and pLATV-algorithm generate visually the best results, while the result of the SA-TV-algorithm seems in some parts over-smoothed. For example, the very left tower in the SA-TV-reconstruction is completely vanished. This object is in the other restorations still visible. For large standard deviations, i.e. $\sigma = 0.3$, we observe from Table 7.10 that the SA-TV method performs clearly worse than the other methods, while the pAPS-algorithm usually has larger PSNR and the LATV- and pLATV-algorithm have larger MSSIM. Hence, whenever the noise-level is too large and details are considerably lost due to noise, the locally adaptive methods are not able to improve the restoration quality.

7.4 Deblurring and Gaussian noise removal

The performance of the algorithms for restoring images corrupted by Gaussian blur with blurring kernel of size 5×5 pixels and standard deviation 10 and additive Gaussian noise with standard deviation σ is reported in Table 7.11. Here we observe that the LATV- as well as the pLATV-algorithm outperform the SA-TV-algorithm for nearly any example. This observation is also clearly visible in Figure 7.5, where the SA-TV-algorithm produces a still blurred output. The pAPS-algorithm generates very similar reconstructions as the LATV- and pLATV-algorithm, which is also reflected by similar PSNR and MSSIM. Similarly as before, the pAPS-algorithm performs best when $\sigma = 0.3$, while for smaller σ the LATV-algorithm has always the best PSNR.

7.5 Impulse noise removal

Since it turns out that the LATV- and pLATV-algorithm produce nearly the same output, here, we compare only our pAPS- and pLATV-algorithm for L^1-TV minimization, with the SA-TV method introduced in [50]. An approximate solution of the minimization problem in the pLATV-algorithm is solved by the semi-smooth Newton method described in Appendix A. For the SA-TV method we use the parameters suggested in [50] and hence $\mathcal{I}_{i,j} = \Omega^{\omega}_{i,j}$. Moreover, we set $\lambda_0 = 0.2$ in our experiments which seems sufficiently small. In Table 7.12

Table 7.8 *PSNR and MSSIM of the reconstruction of the cameraman-image corrupted by Gaussian white noise with standard deviation $\sigma = 0.1$ via the LATV- and pLATV-algorithm with different α_0 and via the SA-TV-algorithm with different λ_0. In the LATV- and pLATV-algorithm we use $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$ with window-size 11×11 pixels in the interior and we set $p_0 = \frac{1}{2}$.*

	SA-TV		LATV		pLATV		pLATV with Algorithm 1	
α_0/λ_0	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
1	27.82	0.8155	27.44	0.8258	27.37	0.8260	27.37	0.8168
10^{-1}	27.77	0.8123	27.59	0.8211	27.41	0.8189	27.38	0.8166
10^{-2}	27.71	0.8107	27.39	0.8167	27.37	0.8167	27.37	0.8167
10^{-3}	27.42	0.8007	27.40	0.8167	27.38	0.8168	27.38	0.8168
10^{-4}	27.56	0.7792	27.40	0.8168	27.38	0.8168	27.38	0.8168
$\text{PSNR}_{\text{diff}}$	0.39646		0.20257		0.044473		0.012704	
$\text{MSSIM}_{\text{diff}}$	0.036322		0.0091312		0.0092963		0.00019843	

Table 7.9 *PSNR and MSSIM of the reconstruction of different images corrupted by Gaussian white noise with standard deviation σ via the pLATV-algorithm with $\alpha_0 = 10^{-4}$.*

		pLATV with $\mathcal{I} = \tilde{\Omega}$		pLATV with $\mathcal{I} = \Omega$	
Image	σ	PSNR	MSSIM	PSNR	MSSIM
cameraman	0.3	22.47	0.6807	22.47	0.6809
	0.1	27.38	0.8168	27.37	0.8165
	0.05	30.91	0.8875	30.92	0.8875
	0.01	40.69	0.9735	40.68	0.9735
lena	0.3	22.31	0.5947	22.30	0.5950
	0.1	26.85	0.7447	26.87	0.7448
	0.05	30.15	0.8301	30.15	0.8300
	0.01	39.69	0.9699	39.68	0.9699

Table 7.10 *PSNR- and MSSIM-values of the reconstruction of different images corrupted by Gaussian white noise with standard deviation σ via pAPS-, LATV-, pLATV-algorithm with $\alpha_0 = 10^{-4}$ and SA-TV-algorithm with $\lambda_0 = 10^{-4}$. In the LATV- and pLATV-algorithm we use $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$ with window-size 11×11 pixels in the interior and we set $p_0 = \frac{1}{2}$.*

		pAPS (scalar α)		SA-TV		LATV		pLATV	
Image	σ	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
phantom	0.3	19.84	0.7989	19.83	0.8319	**20.35**	0.8411	20.31	**0.8432**
	0.1	28.97	0.9644	28.97	0.9648	**29.50**	**0.9680**	**29.50**	**0.9680**
	0.05	34.97	0.9887	33.77	0.9867	**35.51**	**0.9882**	**35.51**	**0.9882**
	0.01	48.88	**0.9994**	47.38	0.9987	49.46	0.9993	**49.53**	0.9993
cameraman	0.3	**22.62**	**0.6911**	22.03	0.6806	22.47	0.6807	22.47	0.6807
	0.1	27.31	0.8109	**27.56**	0.7792	27.40	0.8168	27.38	**0.8168**
	0.05	30.75	0.8788	**31.60**	**0.8929**	30.95	0.8878	30.91	0.8875
	0.01	40.51	0.9731	**40.92**	0.9649	40.73	**0.9737**	40.69	0.9735
barbara	0.3	**21.22**	0.5022	19.78	0.4470	21.05	**0.5032**	21.05	**0.5032**
	0.1	24.70	0.7145	**25.53**	**0.7292**	24.93	0.7278	24.93	0.7278
	0.05	28.22	0.8514	**29.94**	**0.8801**	28.49	0.8584	28.49	0.8584
	0.01	38.91	0.9791	**40.56**	**0.9809**	39.08	0.9788	39.08	0.9788
lena	0.3	**22.42**	0.5930	21.09	0.5474	22.33	**0.5951**	22.31	0.5947
	0.1	26.84	0.7393	**27.31**	**0.7528**	26.85	0.7447	26.85	0.7447
	0.05	30.06	0.8261	**30.92**	**0.8385**	30.16	0.8307	30.15	0.8301
	0.01	39.62	0.9685	**39.81**	0.9660	39.76	**0.9708**	39.69	0.9699

and Table 7.13 we report on the results obtained by the pAPS-, SA-TV-, and pLATV-algorithm for restoring images corrupted by salt-and-pepper noise or random-valued impulse noise, respectively. While the pAPS- and pLATV-algorithm produce quite similar restorations for both type of noises, the SA-TV algorithm seems to be outperformed in most examples. For example, in Figure 7.6 we observe that the pAPS- and pLATV-algorithm remove the noise considerable while the solution of the SA-TV method still contains noise. On the contrary for the removal of random-valued impulse noise in Figure 7.7 we see that all three methods produce similar restorations.

Table 7.11 *PSNR- and MSSIM-values of the reconstruction of different images corrupted by Gaussian blur (blurring kernel of size 5×5 pixels with standard deviation 10) and additive Gaussian noise with standard deviation σ via pAPS-, LATV-, pLATV-algorithm with $\alpha_0 = 10^{-2}$ and SA-TV-algorithm with $\lambda_0 = 10^{-4}$. In the LATV- and pLATV-algorithm we use $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$ with window-size 11×11 pixels in the interior and set $p_0 = \frac{1}{2}$.*

		pAPS (scalar α)		SA-TV		LATV		pLATV	
Image	σ	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
phantom	0.3	16.23	0.6958	15.65	0.6632	**16.32**	0.6995	16.31	**0.6997**
	0.1	17.86	0.7775	17.31	0.7442	**17.98**	0.7914	17.97	**0.7923**
	0.05	18.89	0.7784	19.20	0.8193	**19.49**	**0.8343**	19.39	0.8279
cameraman	0.3	**21.04**	**0.6410**	19.26	0.5990	20.86	0.6272	20.86	0.6272
	0.1	23.11	**0.7175**	22.64	0.6957	**23.17**	0.7157	23.15	0.7156
	0.05	24.14	0.7562	23.75	0.7393	**24.22**	**0.7573**	24.21	0.7570
barbara	0.3	**20.58**	**0.4556**	18.95	0.4314	20.42	0.4517	20.42	0.4515
	0.1	**22.16**	0.5589	22.09	**0.5687**	**22.16**	0.5597	**22.16**	0.5597
	0.05	22.87	0.6245	22.88	0.6268	**22.92**	**0.6273**	22.90	0.6255
lena	0.3	**21.75**	**0.5542**	20.10	0.5278	21.71	0.5529	21.69	0.5528
	0.1	24.44	0.6496	24.39	**0.6574**	**24.50**	0.6514	24.49	0.6510
	0.05	25.83	0.7047	25.81	**0.7091**	**25.92**	0.7066	25.91	0.7062

Table 7.12 *PSNR- and MSSIM-values of the reconstruction of different images corrupted by salt-and-pepper noise with $r_1 = r_2$ via pAPS-, pLATV-algorithm with $\alpha_0 = 10^{-2}$ and SA-TV-algorithm with $\lambda_0 = 0.2$ and window-size 21×21. In the pLATV-algorithm we use $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$ with window-size 11×11 pixels in the interior and we set $p_0 = \frac{1}{2}$.*

		pAPS (scalar α)			SA-TV			pLATV		
Image	$r_1 = r_2$	PSNR	MSSIM	MAE	PSNR	MSSIM	MAE	PSNR	MSSIM	MAE
phantom	0.3	14.48	0.7040	**0.0519**	**15.28**	0.6540	0.0605	**14.50**	**0.7053**	**0.0519**
	0.1	18.39	0.8412	0.0214	**19.57**	**0.8703**	**0.0196**	18.63	0.8610	0.0202
	0.05	21.61	0.9257	**0.0103**	**22.81**	**0.9362**	**0.0103**	21.55	**0.9327**	0.0104
cameraman	0.3	**21.60**	0.7269	**0.0343**	21.34	0.6871	0.0390	21.59	**0.7271**	0.0344
	0.1	25.49	0.8822	0.0155	**25.80**	0.8774	0.0157	25.57	**0.8844**	**0.0154**
	0.05	**28.80**	**0.9389**	**0.0087**	28.50	0.9251	0.0095	27.99	0.9263	0.0095
barbara	0.3	**21.56**	0.6242	**0.0486**	20.54	0.5889	0.0537	21.51	**0.6253**	0.0488
	0.1	25.49	0.8729	0.0211	25.27	0.8650	**0.0202**	**25.81**	**0.8759**	0.0208
	0.05	**28.46**	**0.9338**	0.0118	27.90	0.9313	**0.0110**	28.34	0.9325	0.0121
lena	0.3	23.29	0.6807	0.0360	22.61	0.6397	0.0404	**23.32**	**0.6811**	**0.0359**
	0.1	27.60	0.8508	0.0151	27.78	0.8459	0.0152	**27.99**	**0.8530**	**0.0148**
	0.05	29.45	**0.8946**	**0.0096**	**29.74**	0.8863	0.0100	29.53	0.8931	0.0097

Table 7.13 *PSNR- and MSSIM-values of the reconstruction of different images corrupted by random-valued impulse noise via pAPS-, pLATV-algorithm with $\alpha_0 = 10^{-2}$ and SA-TV-algorithm with $\lambda_0 = 0.2$ and window-size 21×21. In the pLATV-algorithm we use $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$ with window-size 11×11 pixels in the interior and we set $p_0 = \frac{1}{2}$ in the pLATV-algorithm.*

		pAPS (scalar α)			SA-TV			pLATV		
Image	r	PSNR	MSSIM	MAE	PSNR	MSSIM	MAE	PSNR	MSSIM	MAE
phantom	0.3	17.83	0.8120	**0.0317**	**18.68**	0.8012	0.0319	**18.19**	**0.8303**	**0.0305**
	0.1	22.46	0.9273	0.0113	**23.83**	0.9278	**0.0100**	22.58	**0.9328**	0.0112
	0.05	25.55	0.9636	0.0058	**26.56**	0.9642	**0.0054**	25.45	**0.9665**	0.0057
cameraman	0.3	**24.87**	**0.8337**	**0.0213**	23.48	0.7583	0.0237	24.19	0.7887	0.0234
	0.1	**29.33**	**0.9359**	**0.0087**	27.72	0.9087	0.0089	28.60	0.9204	0.0093
	0.05	**31.46**	**0.9603**	0.0053	30.53	0.9478	**0.0052**	30.84	0.9442	0.0058
barbara	0.3	**24.24**	**0.8040**	0.0301	23.96	0.7977	**0.0280**	**24.24**	0.7992	0.0302
	0.1	**29.20**	**0.9355**	0.0118	28.60	0.9327	**0.0101**	28.91	0.9305	0.0120
	0.05	**31.95**	**0.9650**	0.0065	30.65	0.9578	**0.0059**	31.85	0.9640	0.0066
lena	0.3	**26.80**	**0.8124**	**0.0205**	24.74	0.7560	0.0236	26.63	0.8082	0.0208
	0.1	**30.34**	**0.8965**	**0.0092**	28.96	0.8833	0.0095	30.07	0.8918	0.0095
	0.05	**31.36**	**0.9189**	0.0062	30.42	0.9180	**0.0059**	31.08	0.9159	0.0063

8 Conclusion

For L^1-TV and L^2-TV minimization including convolution type of problems automatic parameter selection algorithms for scalar and locally dependent weights α are presented. In particular, we introduce the APS- and pAPS-algorithm for automatically determining a suitable scalar regularization parameter. While for the APS-algorithm its convergence only under some assumptions is shown, the pAPS-algorithm is guaranteed to converge always. Besides the general applicability of these two algorithms they also possess a fast numerical convergence in practice.

(a) pAPS (PSNR: 27.31; MSSIM: 0.8109)

(b) SA-TV (PSNR: 27.56; MSSIM: 0.7792)

(c) LATV (PSNR: 27.40; MSSIM: 0.8170)

(d) pLATV (PSNR: 27.38; MSSIM: 0.8171)

Fig. 7.4 *Reconstruction of the cameraman-image corrupted by Gaussian white noise with* $\sigma = 0.1$.

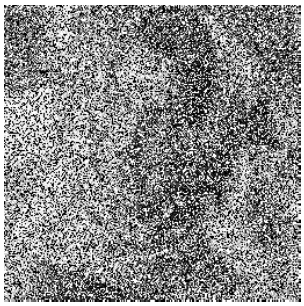
(a) pAPS (PSNR: 23.11; MSSIM: 0.7175)

(b) SA-TV (PSNR: 22.64; MSSIM: 0.6957)

(c) LATV (PSNR: 23.17; MSSIM: 0.7160)

(d) pLATV (PSNR: 23.15; MSSIM: 0.7160)

Fig. 7.5 *Reconstruction of the cameraman-image corrupted by Gaussian white noise with* $\sigma = 0.1$ *and Gaussian blur.*



(a) noisy image

(b) pAPS (PSNR: 21.56; MSSIM: 0.6242)

(c) SA-TV (PSNR: 20.54; MSSIM: 0.5889)

(d) pLATV (PSNR: 21.51; MSSIM: 0.6253)

Fig. 7.6 *Reconstruction of the barbara-image corrupted by salt-and-pepper noise with* $r_1 = r_2 = 0.3$.

(a) noisy image

(b) pAPS (PSNR: 24.24; MSSIM: 0.8040)

(c) SA-TV (PSNR: 23.96; MSSIM: 0.7977)

(d) pLATV (PSNR: 24.24; MSSIM: 0.7992)

Fig. 7.7 *Reconstruction of the barbara-image corrupted by random-valued impulse noise with* $r = 0.3$.

In order to treat homogeneous regions differently than fine features in images, which promises a better reconstruction, algorithms for automatically computing locally adapted weights α are proposed. These methods are much more stable with respect to the initial α_0 than the state-of-the-art SA-TV method. Moreover, while in the SA-TV-algorithm the initial $\lambda_0 > 0$ has to be chosen sufficiently small, in our proposed methods any arbitrary $\alpha_0 > 0$ is allowed. Hence the LATV- and pLATV-algorithm are much more flexible with respect to the initialization. By numerical experiments it is shown that the reconstructions obtained by the newly introduced algorithms are similar with respect to image quality measure to the restorations obtained by the SA-TV algorithm. In the case of Gaussian noise removal (including deblurring) for sufficiently small noise-levels reconstructions obtained by locally varying weights seem to be qualitatively better than results with scalar parameters. On the contrary for removing impulse noise a spatially varying α or λ is in general not always improving the restoration quality.

For computing a minimizer of the respective multiscale total variation model we present first and second order methods and show their convergence to a respective minimizer.

Acknowledgements The author would like to thank M. Monserrat Rincon-Camacho for providing the spatially adaptive parameter selection code for the L^1-TV model of [50].

A Semi-smooth Newton Method for Solving (6.8)

A semi-smooth Newton algorithm for solving (6.8) can be derived similar as in [50] by means of vector-valued variables. Therefore let $u^h \in \mathbb{R}^N$, $q^h \in \mathbb{R}^N$, $\alpha^h \in \mathbb{R}^N$, $g^h \in \mathbb{R}^N$ where $N = N_1 N_2$, denote the discrete image intensity, the dual variable, the spatially dependent regularization parameter, and the observed data vector, respectively. Correspondingly we define $\nabla^h \in \mathbb{R}^{2N\times N}$ as the discrete gradient operator, $\Delta^h \in \mathbb{R}^{N\times N}$ as the discrete Laplace operator, $T^h \in \mathbb{R}^{N\times N}$ as the discrete operator, and $(T^h)^t$ as the transpose of T^h. Here $|\cdot|$, $\max\{\cdot,\cdot\}$, and $\operatorname{sign}(\cdot)$ are understood for vectors in a componentwise sense. Moreover, we use $[|v^h|]_i = \sqrt{(v_i^h)^2 + (v_{i+M}^h)^2}$ for $1 \le i \le N$.

For solving (6.8) in every step of our Newton method we need to solve

$$\begin{pmatrix} A_k^h & -D^h(m_{\beta_k}) & 0 \\ -\frac{1}{\beta+\mu}(T^h)^t T^h + \kappa\Delta^h & -\frac{\mu}{\mu+\beta}(T^h)^t & -(\nabla^h)^t \\ B_k^h & 0 & D^h(m_{\gamma_k}) \end{pmatrix} \begin{pmatrix} \delta_u \\ \delta_v \\ \delta_q \end{pmatrix} = \begin{pmatrix} -\mathfrak{F}_1^k \\ -\mathfrak{F}_2^k \\ -\mathfrak{F}_3^k \end{pmatrix} \tag{A.1}$$

where

$$A_k^h = [D^h(e_N) - D^h(v_k^h)\chi_{\mathcal{A}_{\beta_k}} D^h(\operatorname{sign}(T^h u_k^h - g^h))]T^h,$$

$$B_k^h = [-D^h((\alpha^h,\alpha^h)^t)D^h(q_k^h)\chi_{\mathcal{A}_{\gamma_k}} D^h(m_{\gamma_k})^{-1} M^h(\nabla^h u_k^h)]\nabla^h,$$

$$\mathfrak{F}_1^k = Tu_k^h - g^h - D^h(m_{\beta_k})v_k^h,$$

$$\mathfrak{F}_2^k = -(\nabla^h)^t q_k^h + \kappa\Delta^h u_k^h - \frac{1}{\beta+\mu}(T^h)^t(T^h u_k^h - g^h) - \frac{\mu}{\mu+\beta}(T^h)^t v_k^h,$$

$$\mathfrak{F}_3^k = -D^h((\alpha^h,\alpha^h)^t)\nabla^h u_k^h + D^h(m_{\gamma_k})q_k^h,$$

$e_N \in \mathbb{R}^N$ is the identity vector, $D^h(v)$ is a diagonal matrix with the vector v in its diagonal, $m_{\beta_k} = \max\{\beta, |T^h u_k^h - g^h|\}$, $m_{\gamma_k} = \max\{\gamma\alpha^h, [|\nabla^h u_k^h|]\}$,

$$\chi_{\mathcal{A}_{\beta_k}} = D^h(t_{\beta_k}) \quad \text{with} \quad ((t_{\beta_k})_i = \begin{cases} 0 & \text{if } (m_{\beta_k})_i = \beta, \\ 1 & \text{else;} \end{cases}$$

$$\chi_{\mathcal{A}_{\gamma_k}} = D^h(t_{\gamma_k}) \quad \text{with} \quad ((t_{\gamma_k})_i = \begin{cases} 0 & \text{if } (m_{\gamma_k})_i = \gamma(\alpha^h)_i, \\ 1 & \text{else;} \end{cases}$$

$$M^h(v) = \begin{pmatrix} D^h(v_x) & D^h(v_y) \\ D^h(v_x) & D^h(v_y) \end{pmatrix} \quad \text{with} \quad v = (v_x, v_y)^t \in \mathbb{R}^{2N}.$$

Since the diagonal matrices $D^h(t_{\beta_k})$ and $D^h(t_{\gamma_k})$ are invertible, we eliminate δ_v and δ_q from (A.1), which leads to the following resulting system

$$H_k \delta_u = f_k$$

where

$$H_k := \frac{1}{\beta+\mu}(T^h)^t T^h - \kappa\Delta^h + \frac{\mu}{\mu+\beta}(T^h)^t D^h(m_{\beta_k})^{-1} A_k^h + (\nabla^h)^t D^h(m_{\gamma_k})^{-1}(-B_k^h),$$

$$f_k := \mathfrak{F}_2^k - \frac{\mu}{\mu+\beta}(T^h)^t D^h(m_{\beta_k})^{-1}\mathfrak{F}_1^k + (\nabla^h)^* D^h(m_{\gamma_k})^{-1}\mathfrak{F}_2^k.$$

In general B_k^h and hence H_k is not symmetric. In [51] it is shown that the matrix B_k^h at the solution $(u_k^h, v_k^h, q_k^h) = (\bar{u}, \bar{v}, \bar{q})$ is positive definite whenever

$$[|q_k^h|]_i \le (\alpha^h)_i \quad \text{and} \quad (|v_k^h|)_i \le 1 \tag{A.2}$$

for $i = 1, \dots, N$.

In case these two inequalities are not satisfied we project q_k^h and v_k^h onto their feasible set, i.e., $((q_k^h)_i, (q_k^h)_{i+N})$ is set to $(\alpha^h)_i \max\{(\alpha^h)_i, [|q_k^h|]_i\}^{-1}((q_k^h)_i, (q_k^h)_{i+N})$ and $(v_k^h)_i$ is replaced by $\max\{1, (|v_k^h|)_i\}(v_k^h)_i$. Then the modified system matrix, denoted by H_k^+ is positive definite; see [30]. As pointed out in [50] we may use $H_k^+ + \varepsilon_k D^h(e_N)$ with $\kappa = 0$, $\varepsilon_k > 0$ and $\varepsilon_k \downarrow 0$ as $k \to \infty$ instead of $\kappa > 0$ to obtain a positive definite matrix. Then our semi-smooth Newton solver may be written as in [50]:

Semi-smooth Newton method: Initialize $(u_0^h, q_0^h) \in \mathbb{R}^N \times \mathbb{R}^{2N}$ and set $k := 0$.

1. Determine the active sets $\chi_{\mathcal{A}_{\beta_k}} \in \mathbb{R}^{N\times N}$ and $\chi_{\mathcal{A}_{\gamma_k}} \in \mathbb{R}^{2N\times 2N}$
2. If (A.2) is not satisfied, then compute H_k^+; otherwise set $H_k^+ := H_k$.
3. Solve $H_k^+ \delta_u = f_k$ for δ_u.
4. Compute δ_q by using δ_u.
5. Update $u_{k+1}^h := u_k^h + \delta_u$ and $q_{k+1}^h := q_k^h + \delta_q$.
6. Stop or set $k := k+1$ and continue with step 1).

This algorithm converges at a superlinear rate, which follows from standard theory; see [47,51].

In our experiments we always choose $\kappa = 0$, $\beta = 10^{-3}$, $\gamma = 10^{-2}$, and $\mu = 10^{6}$.

References

1. Adams, R.A., Fournier, J.J.F.: Sobolev spaces, *Pure and Applied Mathematics (Amsterdam)*, vol. 140, second edn. Elsevier/Academic Press, Amsterdam (2003)
2. Alliney, S.: A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. Signal Processing, IEEE Transactions on **45**(4), 913–917 (1997)
3. Almansa, A., Ballester, C., Caselles, V., Haro, G.: A TV based restoration model with local constraints. J. Sci. Comput. **34**(3), 209–236 (2008)
4. Ambrosio, L., Fusco, N., Pallara, D.: Functions of bounded variation and free discontinuity problems. Oxford Mathematical Monographs. The Clarendon Press, Oxford University Press, New York (2000)
5. Aujol, J.F., Gilboa, G., Chan, T., Osher, S.: Structure-texture image decomposition – modeling, algorithms, and parameter selection. International Journal of Computer Vision **67**(1), 111–136 (2006). DOI 10.1007/s11263-006-4331-z
6. Babacan, S.D., Molina, R., Katsaggelos, A.K.: Parameter estimation in TV image restoration using variational distribution approximation. Image Processing, IEEE Transactions on **17**(3), 326–339 (2008)
7. Bartels, S.: Numerical Methods for Nonlinear Partial Differential Equations, vol. 14. Springer (2015)
8. Bertalmío, M., Caselles, V., Rougé, B., Solé, A.: TV based image restoration with local constraints. Journal of scientific computing **19**(1-3), 95–122 (2003)
9. Blomgren, P., Chan, T.F.: Modular solvers for image restoration problems using the discrepancy principle. Numerical linear algebra with applications **9**(5), 347–358 (2002)
10. Blu, T., Luisier, F.: The SURE-LET approach to image denoising. Image Processing, IEEE Transactions on **16**(11), 2778–2786 (2007)
11. Bovik, A.C.: Handbook of image and video processing. Academic press (2010)
12. Braides, A.: Γ-convergence for beginners, *Oxford Lecture Series in Mathematics and its Applications*, vol. 22. Oxford University Press, Oxford (2002)
13. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sci. **3**(3), 492–526 (2010)
14. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**(2), 490–530 (2005)
15. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. Information Theory, IEEE Transactions on **52**(2), 489–509 (2006)
16. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vision **20**(1-2), 89–97 (2004). Special issue on mathematics and image analysis
17. Chambolle, A., Darbon, J.: On total variation minimization and surface evolution using parametric maximum flows. International journal of computer vision **84**(3), 288–307 (2009)
18. Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. Numer. Math. **76**(2), 167–188 (1997)
19. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision **40**(1), 120–145 (2011)
20. Chan, T.F., Esedoḡlu, S.: Aspects of total variation regularized L^1 function approximation. SIAM J. Appl. Math. **65**(5), 1817–1837 (2005)
21. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. SIAM J. Sci. Comput. **20**(6), 1964–1977 (1999)
22. Chan, T.F., Shen, J., Zhou, H.M.: Total variation wavelet inpainting. Journal of Mathematical imaging and Vision **25**(1), 107–125 (2006)
23. Ciarlet, P.G.: Introduction to numerical linear algebra and optimisation. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge (1989). With the assistance of Bernadette Miara and Jean-Marie Thomas, Translated from the French by A. Buttigieg
24. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**(4), 1168–1200 (electronic) (2005)
25. Darbon, J., Sigelle, M.: A fast and exact algorithm for total variation minimization. In: Pattern recognition and image analysis, pp. 351–359. Springer (2005)
26. Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation. I. Fast and exact optimization. J. Math. Imaging Vision **26**(3), 261–276 (2006)
27. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Comm. Pure Appl. Math. **57**(11), 1413–1457 (2004)
28. Daubechies, I., Teschke, G., Vese, L.: Iteratively solving linear inverse problems under general convex constraints. Inverse Probl. Imaging **1**(1), 29–46 (2007)
29. Dobson, D.C., Vogel, C.R.: Convergence of an iterative method for total variation denoising. SIAM J. Numer. Anal. **34**(5), 1779–1791 (1997)
30. Dong, Y., Hintermüller, M., Neri, M.: An efficient primal-dual method for l^1 tv image restoration. SIAM Journal on Imaging Sciences **2**(4), 1168–1189 (2009)
31. Dong, Y., Hintermüller, M., Rincon-Camacho, M.M.: Automated regularization parameter selection in multi-scale total variation models for image restoration. J. Math. Imaging Vision **40**(1), 82–104 (2011)
32. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. Journal of the american statistical association **90**(432), 1200–1224 (1995)
33. Ekeland, I., Témam, R.: Convex analysis and variational problems, *Classics in Applied Mathematics*, vol. 28, english edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1999). Translated from the French
34. Eldar, Y.C.: Generalized sure for exponential families: Applications to regularization. Signal Processing, IEEE Transactions on **57**(2), 471–481 (2009)
35. Engl, H.W., Grever, W.: Using the L-curve for determining optimal regularization parameters. Numer. Math. **69**(1), 25–31 (1994)
36. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of inverse problems, *Mathematics and its Applications*, vol. 375. Kluwer Academic Publishers Group, Dordrecht (1996)
37. Fornasier, M., Naumova, V., Pereverzyev, S.V.: Parameter choice strategies for multipenalty regularization.

SIAM Journal on Numerical Analysis **52**(4), 1770–1794 (2014)
38. Gilboa, G., Sochen, N., Zeevi, Y.Y.: Texture preserving variational denoising using an adaptive fidelity term. In: Proc. VLsM, vol. 3 (2003)
39. Giryes, R., Elad, M., Eldar, Y.C.: The projected GSURE for automatic parameter tuning in iterative shrinkage methods. Appl. Comput. Harmon. Anal. **30**(3), 407–422 (2011)
40. Giusti, E.: Minimal surfaces and functions of bounded variation, *Monographs in Mathematics*, vol. 80. Birkhäuser Verlag, Basel (1984)
41. Goldstein, T., Osher, S.: The split Bregman method for $L1$-regularized problems. SIAM J. Imaging Sci. **2**(2), 323–343 (2009)
42. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979)
43. Hanke, M.: Limitations of the L-curve method in ill-posed problems. BIT **36**(2), 287–301 (1996)
44. Hansen, P.C.: Analysis of discrete ill-posed problems by means of the L-curve. SIAM Rev. **34**(4), 561–580 (1992)
45. Hansen, P.C., O'Leary, D.P.: The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J. Sci. Comput. **14**(6), 1487–1503 (1993)
46. He, C., Hu, C., Zhang, W., Shi, B.: A fast adaptive parameter estimation for total variation image restoration. Image Processing, IEEE Transactions on **23**(12), 4954–4967 (2014)
47. Hintermüller, M., Kunisch, K.: Total bounded variation regularization as a bilaterally constrained optimization problem. SIAM Journal on Applied Mathematics **64**(4), 1311–1333 (2004)
48. Hintermüller, M., Langer, A.: Subspace correction methods for a class of nonsmooth and nonadditive convex variational problems with mixed L^1/L^2 data-fidelity in image processing. SIAM J. Imaging Sci. **6**(4), 2134–2173 (2013)
49. Hintermüller, M., Langer, A.: Adaptive regularization for parseval frames in image processing. SFB-Report No. 2014-014 p. 12 (2014)
50. Hintermüller, M., Rincon-Camacho, M.M.: Expected absolute value estimators for a spatially adapted regularization parameter choice rule in L^1-TV-based image restoration. Inverse Problems **26**(8), 085,005, 30 (2010)
51. Hintermüller, M., Stadler, G.: An infeasible primal-dual algorithm for total bounded variation-based inf-convolution-type image restoration. SIAM J. Sci. Comput. **28**(1), 1–23 (2006)
52. Kindermann, S., Osher, S., Jones, P.W.: Deblurring and denoising of images by nonlocal functionals. Multiscale Model. Simul. **4**(4), 1091–1115 (electronic) (2005)
53. Langer, A.: Subspace correction and domain decomposition methods for total variation minimization. Ph.D. thesis, Johannes Kepler Universtät Linz (2011)
54. Li, F., Ng, M.K., Shen, C.: Multiplicative noise removal with spatially varying regularization parameters. SIAM J. Imaging Sci. **3**(1), 1–20 (2010)
55. Liao, H., Li, F., Ng, M.K.: Selection of regularization parameter in total variation image restoration. J. Opt. Soc. Amer. A **26**(11), 2311–2320 (2009)
56. Lin, Y., Wohlberg, B., Guo, H.: UPRE method for total variation parameter selection. Signal Processing **90**(8), 2546–2551 (2010)
57. Mallows, C.L.: Some comments on C_P. Technometrics **15**(4), 661–675 (1973)
58. Morozov, V.A.: Methods for solving incorrectly posed problems. Springer-Verlag, New York (1984). Translated from the Russian by A. B. Aries, Translation edited by Z. Nashed
59. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. **42**(5), 577–685 (1989)
60. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1, Ser. A), 127–152 (2005)
61. Ng, M.K., Weiss, P., Yuan, X.: Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. SIAM journal on Scientific Computing **32**(5), 2710–2736 (2010)
62. Nikolova, M.: Minimizers of cost-functions involving non-smooth data-fidelity terms. Application to the processing of outliers. SIAM J. Numer. Anal. **40**(3), 965–994 (electronic) (2002)
63. Nikolova, M.: A variational approach to remove outliers and impulse noise. Journal of Mathematical Imaging and Vision **20**(1-2), 99–120 (2004)
64. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. Multiscale Model. Simul. **4**(2), 460–489 (electronic) (2005)
65. Papafitsoros, K., Schönlieb, C.B.: A combined first and second order variational approach for image reconstruction. Journal of mathematical imaging and vision **48**(2), 308–338 (2014)
66. Rockafellar, R.T.: Convex analysis. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J. (1970)
67. Rudin, L.I., Osher, S.: Total variation based image restoration with free local constraints. In: Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, vol. 1, pp. 31–35. IEEE (1994)
68. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1), 259–268 (1992)
69. Stein, C.M.: Estimation of the mean of a multivariate normal distribution. The annals of Statistics pp. 1135–1151 (1981)
70. Strong, D.M., Blomgren, P., Chan, T.F.: Spatially adaptive local-feature-driven total variation minimizing image restoration. In: Optical Science, Engineering and Instrumentation'97, pp. 222–233. International Society for Optics and Photonics (1997)
71. Strong, D.M., Chan, T.F.: Spatially and scale adaptive total variation based regularization and anisotropic diffusion in image processing. In: Diusion in Image Processing, UCLA Math Department CAM Report. Citeseer (1996)
72. Sutour, C., Deledalle, C.A., Aujol, J.F.: Adaptive regularization of the nl-means: Application to image and video denoising. Image Processing, IEEE Transactions on **23**(8), 3506–3521 (2014)
73. Tadmor, E., Nezzar, S., Vese, L.: A multiscale image representation using hierarchical (BV, L^2) decompositions. Multiscale Model. Simul. **2**(4), 554–579 (electronic) (2004)
74. Tadmor, E., Nezzar, S., Vese, L.: Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation. Commun. Math. Sci. **6**(2), 281–307 (2008)
75. Tikhonov, A.N., Arsenin, V.Y.: Solutions of ill-posed problems. Vh Winston (1977)
76. Vogel, C.R.: Non-convergence of the L-curve regularization parameter selection method. Inverse Problems **12**(4), 535–547 (1996)

77. Vogel, C.R.: Computational methods for inverse problems, vol. 23. Siam (2002)
78. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on **13**(4), 600–612 (2004)
79. Weiss, P., Blanc-Féraud, L., Aubert, G.: Efficient schemes for total variation minimization under constraints in image processing. SIAM J. Sci. Comput. **31**(3), 2047–2080 (2009)
80. Wen, Y.W., Chan, R.H.: Parameter selection for total-variation-based image restoration using discrepancy principle. IEEE Transactions on Image Processing **21**(4), 1770–1781 (2012)
81. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report pp. 08–34 (2008)