

# Automated Parameter Selection for Total Variation Minimization in Image Restoration

Andreas Langer<sup>1</sup>

Received: 21 January 2016 / Accepted: 5 July 2016 / Published online: 22 July 2016  
© Springer Science+Business Media New York 2016

**Abstract** Algorithms for automatically selecting a scalar or locally varying regularization parameter for total variation models with an  $L^{\tau}$ -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 multiscale total variation minimization problem. For solving the constituted multiscale 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** Locally dependent regularization parameter · Automated parameter selection · 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 by either 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  $\eta$  is oscillatory with zero mean and standard deviation  $\sigma$ , 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

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

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

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

where  $\rho$  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 [41]. A good approximation of  $\hat{u}$  may be obtained by solving a minimization problem of the type

✉ Andreas Langer  
andreas.langer@mathematik.uni-stuttgart.de

<sup>1</sup> Institute for Applied Analysis and Numerical Simulation,  
University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart,  
Germany

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

where  $\mathcal{H}(\cdot; g)$  represents a data fidelity term, which enforces the consistency between the recovered and measured images,  $\mathcal{R}$  is an appropriate filter or regularization term, which prevents overfitting, 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 [81] for image denoising. Hence, here and in the remaining of the paper we choose  $\mathcal{R}(u) = \int_{\Omega} |Du|$ , where  $\int_{\Omega} |Du|$  denotes the total variation of  $u$  in  $\Omega$ ; see [3,46] for more details. However, we note that other regularization terms, such as the total generalized variation [13], the nonlocal total variation [60], the Mumford–Shah regularizer [71], or higher order regularizers (see e.g. [77] and references therein) might be used as well.

### 1.1 Choice of the Fidelity Term

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 [18–20,24,28–30,32,35,47,72,76,91,93]. 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

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

where  $BV(\Omega)$  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 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 [1,74,75], uses a nonsmooth  $L^1$ -data fidelity term instead of the  $L^2$ -data fidelity term in (1.4), i.e., one considers

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

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

$$\min_{u \in BV(\Omega)} \left\{ \mathcal{J}_{\tau}(u; g) := \mathcal{H}_{\tau}(u; g) + \alpha \int_{\Omega} |Du| \right\} \tag{1.6}$$

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

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

where  $\lambda = \frac{1}{\alpha} > 0$ . Other and different fidelity terms have been considered in connection with other type of noise models, as Poisson noise [64], multiplicative noise [4], and Rician noise [43]. For images which are simultaneously contaminated by Gaussian and impulse noise [15], a combined  $L^1 - L^2$ -data fidelity term has been recently suggested and demonstrated to work satisfactorily [53]. 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.

### 1.2 Choice of the Scalar Regularization Parameter

For the reconstruction of such images, the proper choice of  $\alpha$  in (1.6) and  $\lambda$  in (1.7) is delicate; cf. Fig. 1. In particular, large  $\alpha$  and small  $\lambda$ , which lead to an oversmoothed reconstruction, not only remove noise but also eliminate details in images. On the other hand, small  $\alpha$  and large  $\lambda$  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  $\alpha$  and, respectively,  $\lambda$  such that a good compromise of the aforementioned effects is made. There are several ways of how to select  $\alpha$  in (1.6) and equivalently  $\lambda$  in (1.7), such as manually by the trial-and-error method, the unbiased predictive risk estimator method (UPRE) [67,69], the Stein unbiased risk estimator method (SURE) [10,38,82] and its generalizations [34,40,45], the generalized cross-validation method (GCV) [48,66,67,78], the L-curve method [49,50], the discrepancy principle [70], and the variational Bayes’ approach [6]. Further parameter selection methods for general inverse problems can be found for example in [41,42,88,89].

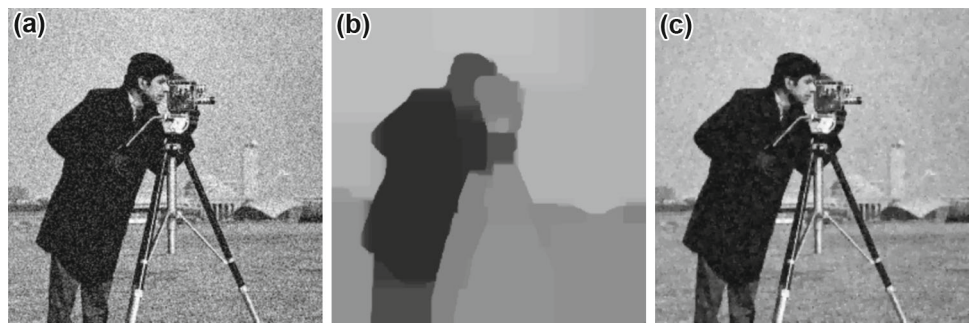
Based on a training set of pairs  $(g_k, \hat{u}_k)$ , for  $k = 1, 2, \dots, N \in \mathbb{N}$ , where  $g_k$  is the noisy observation and  $\hat{u}_k$  represents the original image, for example in [16,33,61] bilevel optimization approaches have been presented to compute suitable scalar regularization parameters of the corresponding image model. Since in our setting we do not have a training set given, these approaches are not applicable here.

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

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

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  $\Omega$ ; see Sect. 3 for more details. Note that

**Fig. 1** Reconstruction of an image corrupted by Gaussian white noise with a “relatively” large parameter  $\alpha$  in (b) and a “relatively” small parameter  $\alpha$  in (c). **a** Noisy image, **b** oversmoothed reconstruction, and **c** overfitted reconstruction



here we assume to know a priori the noise level. In real applications, this means that possibly in a first step a noise estimation has to be performed before the discrepancy principle may be used. However, in general it is easier to estimate the noise level than the regularization parameter [18].

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 Sect. 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, 18, 51, 92] and references therein, while not so much attention has been given to the case  $\tau = 1$ , see for example [73, 91].

### 1.3 Spatially Adaptive Parameter

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  $\alpha$  should be small, or  $\lambda$  should be large, in parts with small features in order to preserve the details. On the contrary,  $\alpha$  should be large, or  $\lambda$  should be small, in homogeneous parts to remove noise considerably. With such a choice of a spatially varying weight, we expect better reconstructions than with a globally constant parameter, as demonstrated for example in [37, 58]. This motivated to consider multiscale total variation models with spatially varying parameters initially suggested in [80]. The multiscale 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

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

and in the sequel we refer to (1.9) and (1.10) as the multiscale  $L^\tau$ -TV model.

In [84], the influence of the scale of an image feature on the choice of  $\alpha$  is studied and the obtained observations were later used in [83] to construct an updating scheme of  $\alpha$ . Based on (1.10), in [8] a piecewise constant function  $\lambda$ , where the pieces are defined by a partitioning of the image due to a presegmentation, is determined. In particular, for each segment a scalar  $\lambda_i, i = 1, \dots, \#\text{pieces}$  is computed by Uzawa’s method [27].

Later, it was noticed that stable choices of  $\lambda$ , respectively,  $\alpha$  should incorporate statistical properties of the noise. In this vein, in [2, 37, 44] for the problem (1.10) automated update rules for  $\lambda$  based on statistics of local constraints were proposed. In [44], 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 are utilized. An adaptation of this approach to multiplicative noise can be found in [65]. For convolution type of problems in [2] based on an estimate of the noise variance for each pixel, an automatic updating scheme of  $\lambda$  using Uzawa’s method is created. This approach is improved in [37] 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 [86, 87], 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 [58]. A different approach has been proposed in [85] for image denoising only, where nonlocal means [14] are used to create a nonlocal data fidelity term. While in all these approaches the adjustment of  $\lambda$  relies on the output of  $T$  being a deteriorated image again, in [54] the method of [37] is adjusted to the situation where  $T$  is an orthogonal wavelet transform or Fourier transform. Very recently, also bilevel optimisation approaches are considered for computing spatially adaptive weights [26, 56, 57].

## 1.4 Contribution

Our first contribution of this paper is to present a method which automatically computes the regularization parameter  $\alpha$  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 [18], 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 [18] is shown to converge to a parameter  $\alpha^*$  such that the corresponding minimizer  $u_{\alpha^*}$  of (1.6) is also a solution of (1.8). The proof relies on the nonincrease 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 [18] 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. Utilizing an appropriate update of  $\alpha$ , which is different than the one used in [18], 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 [18] 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 [18] till termination.

Motivated by multiscale total variation minimization, the second contribution of this paper is concerned with the automated selection of a suitable spatially varying  $\alpha$  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  $\alpha$  is fully automatic. Differently to the scalar case, the adjustment of  $\alpha$  is now based on local constraints, similarly as already considered for example in [2, 37, 58]. 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  $\alpha$  which does not need any additional parameter and hence is not similar to Uzawa's method. Moreover, differently to the approaches in [37, 58] 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, our algorithm is even more general than the ones presented in [37, 58].

## 1.5 Outline of the Paper

The remaining of the paper is organized as follows: In Sect. 2, we revisit and discuss the connection between

the constrained minimization problem (1.8) and the unconstrained optimization problem (1.6). Section 3 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 Sect. 4 our new locally adapted total variation algorithm in detail. Algorithms for performing total variation minimization for spatially varying  $\alpha$  are presented in Sect. 5 where also their convergence properties are studied. To demonstrate the performance of the new algorithms, we present in Sect. 6 numerical experiments for image denoising and image deblurring. Finally, in Sect. 7 conclusions are drawn.

## 2 Constrained Versus Unconstrained Minimization Problem

In this section, we discuss 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  $\mathcal{J} : \mathcal{V} \rightarrow \mathbb{R} \cup \{+\infty\}$  is defined as the set

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

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

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

For a convex functional  $\mathcal{J} : \mathcal{V} \rightarrow \mathbb{R} \cup \{+\infty\}$ , 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

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

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  $L^2(\Omega)$ . Moreover,  $g_\Omega$  describes the average value of the function  $g \in L^1(\Omega)$  in  $\Omega$  defined by  $g_\Omega := \frac{1}{|\Omega|} \int_\Omega g(x) \, dx$ .

**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*

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

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

*Proof* For a proof, we refer the reader to [20] and [58].  $\square$

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  $v_\tau |\Omega| \leq \|g - g_\Omega\|_{L^\tau(\Omega)}^\tau$ . 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 [20]. We state the proof for  $\tau = 1$  by noting that 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  $\tilde{u} = u + g_\Omega$ . We consider now the continuous function  $f(s) = \|T(su + g_\Omega) - g\|_{L^1(\Omega)}$  for  $s \in [0, 1]$ . Note that  $f(1) = \|T\tilde{u} - g\|_{L^1(\Omega)} \leq v_1 |\Omega|$  and  $f(0) = \|Tg_\Omega - g\|_{L^1(\Omega)} = \|g - g_\Omega\|_{L^1(\Omega)} \geq v_1 |\Omega|$ , since  $T \cdot 1 = 1$ , and hence there exists some  $s \in [0, 1]$  such that  $f(s) = v_1 |\Omega|$ . Set  $u' = su$  which satisfies  $\|Tu' - g\|_{L^1(\Omega)} = v_1 |\Omega|$  and

$$\int_\Omega |Du'| = s \int_\Omega |Du| \leq \liminf_{n \rightarrow \infty} \int_\Omega |Du_n|,$$

where  $(u_n)_n$  is a minimizing sequence of (1.8). Hence  $u'$  is a solution of (1.8).  $\square$

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  $v_\tau |\Omega| \leq \|g - g_\Omega\|_{L^\tau(\Omega)}^\tau$ . 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 [20, Prop. 2.1]. By similar arguments, one can show the statement for  $\tau = 1$ , which we state here. Set  $\mathcal{R}(u) := \int_\Omega |Du|$  and

$$\mathcal{G}(u) = \begin{cases} +\infty & \text{if } \|u - g\|_{L^1(\Omega)} > v_1 |\Omega|, \\ 0 & \text{if } \|u - g\|_{L^1(\Omega)} \leq v_1 |\Omega|. \end{cases}$$

Notice that  $\mathcal{R}$  and  $\mathcal{G}$  are convex l.s.c functions and problem (2.1) is equivalent to  $\min_u \mathcal{R}(u) + \mathcal{G}(Tu)$ . We have  $\text{Dom}(\mathcal{R}) = BV(\Omega) \cap L^2(\Omega)$  and  $\text{Dom}(\mathcal{G}) = \{u \in L^2(\Omega) : \mathcal{G}(u) < +\infty\}$ . Since  $g \in \overline{T \text{Dom}(\mathcal{R})}$ , there exists  $\tilde{u} \in \text{Dom}(\mathcal{R})$  with  $\|T\tilde{u} - g\|_{L^1(\Omega)} \leq v_1 |\Omega|/2$ . As  $T \in \mathcal{L}(L^2(\Omega))$  is continuous,  $\mathcal{G} \circ T$  is continuous at  $\tilde{u}$ . Hence, by [39, Prop. 5.6, p. 26] we obtain

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

for all  $u$ . Further,  $\mathcal{G}$  is continuous at  $T\tilde{u}$ , and hence by [39, Prop. 5.7, p. 27] we have for all  $u$

$$\partial(\mathcal{G} \circ T)(u) = T^* \partial\mathcal{G}(Tu),$$

where  $\partial\mathcal{G}(u) = \{0\}$  if  $\|u - g\|_{L^1(\Omega)} < v_1 |\Omega|$  and  $\partial\mathcal{G}(u) = \{\alpha \partial(\|u - g\|_{L^1(\Omega)}), \alpha \geq 0\}$  if  $\|u - g\|_{L^1(\Omega)} = v_1 |\Omega|$ .

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

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

Since any solution of (1.8) satisfies  $\|Tu - g\|_{L^1(\Omega)} = v_1 |\Omega|$ , 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  $\alpha$  is obviously a solution of (1.8) with  $\|Tu - g\|_{L^1(\Omega)} = v_1 |\Omega|$ . This concludes the proof.  $\square$

Note that  $\|Tu_\alpha - g\|_{L^1(\Omega)}$  is (only) convex with respect to  $Tu_\alpha$ , and hence a minimizer of (1.5) is in general not unique even in the simple case when  $T = I$ , i.e., for two minimizers  $u_\alpha^1$  and  $u_\alpha^2$  in general we have  $Tu_\alpha^1 \neq Tu_\alpha^2$ . On the contrary,  $\|Tu_\alpha - g\|_{L^2(\Omega)}^2$  is strictly convex with respect to  $Tu_\alpha$ , i.e., for two minimizers  $u_\alpha^1$  and  $u_\alpha^2$  of (1.4) we have  $Tu_\alpha^1 = Tu_\alpha^2$ . Moreover, the function  $\alpha \mapsto \mathcal{H}_1(u_\alpha; g)$  is in general not continuous [23], while  $\alpha \mapsto \mathcal{H}_2(u_\alpha; g)$  indeed is continuous [20], where  $u_\alpha$  is a respective minimizer of (1.6). Hence we have the following further properties:

**Lemma 2.1** *Let  $u_\alpha$  be a minimizer of (1.6). Then  $\alpha \mapsto \mathcal{H}_\tau(u_\alpha; g)$  is nondecreasing for  $\tau = 1, 2$ . Moreover,  $\alpha \mapsto \mathcal{H}_2(u_\alpha; g)$  maps  $\mathbb{R}^+$  onto  $[0, \|g - g_\Omega\|_{L^2(\Omega)}^2]$ .*

*Proof* For a proof see [20,23].  $\square$

**Proposition 2.2** *If  $u_{\alpha_i}$  is a minimizer of*

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

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

$$\|u_{\alpha_1} - u_{\alpha_2}\|_{L^2(\Omega)} \leq C \|g - g_\Omega\|_{L^2(\Omega)}$$

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

*Proof* By [7, Lemma10.2], we have

$$\begin{aligned} \frac{1}{\alpha_1} \|u_{\alpha_1} - u_{\alpha_2}\|_{L^2(\Omega)}^2 &\leq \frac{1}{\alpha_1} (\mathcal{E}(u_{\alpha_2}, \alpha_1) - \mathcal{E}(u_{\alpha_1}, \alpha_1)) \\ \frac{1}{\alpha_2} \|u_{\alpha_1} - u_{\alpha_2}\|_{L^2(\Omega)}^2 &\leq \frac{1}{\alpha_2} (\mathcal{E}(u_{\alpha_1}, \alpha_2) - \mathcal{E}(u_{\alpha_2}, \alpha_2)). \end{aligned}$$

Summing up these two inequalities yields

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

which implies

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

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

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

On the other hand, inequality (2.2) implies

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

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

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

The assertion follows then from (2.3) and (2.4).  $\square$

*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 Automated Scalar Parameter Selection

In order to find a suitable regularization parameter  $\alpha > 0$  of the minimization problem (1.6), we consider the corresponding constrained optimization problem (1.8). Throughout the paper, we assume that  $T$  does not annihilate constant function, which guarantees the existence of a minimizer of the considered optimization problems; see Sect. 2. We recall that in the constraint of (1.8) the value  $\mathcal{B}_\tau$  is defined as

$\mathcal{B}_\tau = \frac{\nu_\tau}{\tau}|\Omega|$ , where  $\nu_\tau \in \mathbb{R}$  is a statistical value depending on the underlying noise and possibly on the original image.

#### 3.1 Statistical Characterization of the Noise

Let us characterize the noise corrupting the image in more detail by making similar considerations as in [58, Sect. 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  $\nu_1$  and  $\nu_2$ , respectively. For images contaminated by Gaussian white noise with standard deviation  $\sigma$ , 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  $\nu_1$  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  $\nu_1$  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. [58]. Here  $\nu_1$  seems to be fixed, while actually  $\nu_1$  depends on the true (unknown) image  $\hat{u}$ . In particular, for salt-and-pepper noise the expected absolute value is given by

$$\nu_1(\hat{u}) := r_2 - (r_2 - r_1) \frac{1}{|\Omega|} \int_{\Omega} (T\hat{u})(x) \, dx \tag{3.1}$$

and for random-valued impulse noise we have

$$\nu_1(\hat{u}) := \frac{1}{|\Omega|} \int_{\Omega} r \left( (T\hat{u})(x)^2 - (T\hat{u})(x) + \frac{1}{2} \right) \, dx. \tag{3.2}$$

However, instead of considering the constraint  $\mathcal{H}_\tau(u; g) = \mathcal{B}_\tau(u)$  in (1.8), which results in a quite nonlinear problem, in our numerics we choose a reference image and compute an approximate value  $\mathcal{B}_\tau$ . Since our proposed algorithms are of iterative nature (see APS- and pAPS-algorithm below), it makes sense to choose the current approximation as the reference image, i.e., the reference image changes during the iterations. 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,  $\nu_\tau$  and  $\mathcal{B}_\tau$  are independent of  $\hat{u}$  too. Nevertheless, in order to keep the paper concise, in the sequel instead of  $\nu_\tau$  and  $\mathcal{B}_\tau$  we often write  $\nu_\tau(\bar{u})$  and  $\mathcal{B}_\tau(\bar{u})$ , where  $\bar{u}$  represents a reference image approximating  $\hat{u}$ , even if the values may actually be independent from the image.

#### 3.2 Automated Parameter Selection Strategy

In order to determine a suitable regularization parameter  $\alpha$  in [18], an algorithm for solving the constrained minimization problem (1.8) for  $T = I$  and  $\tau = 2$  is proposed, i.e., in the

presence of Gaussian noise with zero mean and standard deviation  $\sigma$ . This algorithm relies on the fact that  $\alpha \mapsto \mathcal{H}_2(u_\alpha; g)$  is nondecreasing, which leads to the following iterative procedure.

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

1) Compute

$$u_{\alpha_n} \in \arg \min_{u \in BV(\Omega)} \|u - g\|_{L^2(\Omega)}^2 + 2\alpha_n \int_{\Omega} |Du|$$

2) Update  $\alpha_{n+1} := \frac{\sigma \sqrt{|\Omega|}}{\|u_{\alpha_n} - g\|_{L^2(\Omega)}} \alpha_n$ .

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

For the minimization of the optimization problem in step 1) in [18], 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  $n \rightarrow \infty$ ,  $\|u_{\alpha_n} - g\|_{L^2(\Omega)} \rightarrow \sigma \sqrt{|\Omega|}$  and  $u_{\alpha_n}$  converges to the unique solution of (1.8) with  $T = I$  and  $\tau = 2$  [18]. The proof relies on the fact that the function  $\alpha \rightarrow \frac{\|u_\alpha - g\|_{L^2(\Omega)}}{\alpha}$  is nonincreasing. Note that this property does not hold in general for operators  $T \neq I$ .

### 3.2.1 The $p$ -Adaptive Algorithm

We generalize now the CPS-algorithm to optimization problems of the type (1.8) for  $\tau = 1, 2$  and for general operators  $T$ . In order to keep or obtain appropriate convergence properties, we need the following two conditions to be satisfied. Firstly, the function  $\alpha \mapsto \mathcal{H}_\tau(u_\alpha; g)$  has to be monotonic, which is the case due to Lemma 2.1. Secondly, in each iteration  $n$  the parameter  $\alpha_n$  has to be updated such that  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  is monotonic and bounded by  $(\mathcal{B}_\tau(u_{\alpha_n}))_n$ . More precisely, if  $\mathcal{H}_\tau(u_{\alpha_0}; g) \leq \mathcal{B}_\tau(u_{\alpha_0})$ , then there has to exist an  $\alpha_{n+1} \geq \alpha_n$  such that  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau(u_{\alpha_{n+1}})$ , while if  $\mathcal{H}_\tau(u_{\alpha_0}; g) > \mathcal{B}_\tau(u_{\alpha_0})$ , then there has to exist an  $\alpha_{n+1} \leq \alpha_n$  such that  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \geq \mathcal{B}_\tau(u_{\alpha_{n+1}})$ . This holds true by setting in every iteration

$$\alpha_{n+1} := \left( \frac{\mathcal{B}_\tau(u_{\alpha_n})}{\mathcal{H}_\tau(u_{\alpha_n}; g)} \right)^p \alpha_n \tag{3.3}$$

together with an appropriate choice of  $p \geq 0$ . In particular, there exists always a  $p \geq 0$  such that this condition is satisfied.

**Proposition 3.1** Assume  $\|g - g_\Omega\|_{L^\tau(\Omega)} \geq \nu_\tau |\Omega|$  and  $\alpha_{n+1}$  is defined as in (3.3).

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

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

Taking these considerations into account, a generalization of the CPS-algorithm can be formulated as the following  $p$ -adaptive automated parameter selection algorithm:

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

- 1) Compute  $u_{\alpha_n} \in \arg \min_{u \in BV(\Omega)} \mathcal{J}_\tau(u; g)$
- 2) Update  $\alpha_{n+1} := \left( \frac{\mathcal{B}_\tau(u_{\alpha_n})}{\mathcal{H}_\tau(u_{\alpha_n}; g)} \right)^p \alpha_n$  if  $\mathcal{H}_\tau(u_{\alpha_n}; g) > 0$  and continue with step 3). Otherwise increase  $\alpha_n$ , e.g.,  $\alpha_n := 10\alpha_n$ , and go to step 1).
- 3) Compute  $u_{\alpha_{n+1}} \in \arg \min_{u \in BV(\Omega)} \mathcal{J}_\tau(u; g)$
- 4) a) if  $\mathcal{H}_\tau(u_{\alpha_0}; g) \leq \mathcal{B}_\tau(u_{\alpha_0})$ 
  - (i) if  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau(u_{\alpha_{n+1}})$  go to step 5)
  - (ii) if  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) > \mathcal{B}_\tau(u_{\alpha_{n+1}})$ , decrease  $p$ , e.g., set  $p := p/2$ , and go to step 2)
- b) if  $\mathcal{H}_\tau(u_{\alpha_0}; g) > \mathcal{B}_\tau(u_{\alpha_0})$ 
  - (i) if  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \geq \mathcal{B}_\tau(u_{\alpha_{n+1}})$  go to step 5)
  - (ii) if  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) < \mathcal{B}_\tau(u_{\alpha_{n+1}})$ , 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).

Note that due to the dependency of  $\alpha_{n+1}$  on  $p$  a proper  $p$  cannot be explicitly computed, but only iteratively, as in the pAPS-algorithm.

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 Sect. 6 we set  $p_0 = 32$ , which seems large enough to us.

**Proposition 3.2** The pAPS-algorithm generates monotone sequences  $(\alpha_n)_n$  and  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  such that  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  is bounded. Moreover, if  $\mathcal{H}_\tau(u_{\alpha_0}; g) > \mathcal{B}_\tau(u_{\alpha_0})$  or  $\mathcal{B}_\tau(u_\alpha) \leq \frac{1}{\tau} \|g - g_\Omega\|_{L^\tau}^\tau$  for all  $\alpha > 0$ , then  $(\alpha_n)_n$  is also bounded.

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

If  $\mathcal{H}_\tau(u_{\alpha_0}; g) \leq \mathcal{B}_\tau(u_{\alpha_0})$ , due to Lemma 2.1 we have that  $0 < \alpha_n \leq \alpha_{n+1}$  and  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g)$  for all

$n \in \mathbb{N}$  and hence  $(\alpha_n)_n$  and  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  are monotonically increasing. Since there exists  $\mathcal{B}_\tau^* > 0$  such that  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{B}_\tau(u_{\alpha_n}) \leq \mathcal{B}_\tau^*$  for all  $n \in \mathbb{N}$ , see Sect. 3.1,  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  is also bounded. If we additionally assume that  $\mathcal{B}_\tau(u_\alpha) \leq \frac{1}{\tau} \|g - g_\Omega\|_\tau^\tau$  for all  $\alpha > 0$  and we set  $\mathcal{B}_\tau^* := \max_\alpha \mathcal{B}_\tau(u_\alpha)$ , then Theorem 2.2 ensures the existence of an  $\alpha^* \geq 0$  such that  $\mathcal{H}_\tau(u_{\alpha^*}; g) = \mathcal{B}_\tau^*$ . By Lemma 2.1, it follows that  $\alpha_n \leq \alpha^*$  for all  $n \in \mathbb{N}$ , since  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{B}_\tau(u_{\alpha_n}) \leq \mathcal{B}_\tau^*$ . Hence,  $(\alpha_n)_n$  is bounded, which finishes the proof.  $\square$

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 3.2 holds. For constant  $\mathcal{B}_\tau$ , we are even able to argue the convergence of the pAPS-algorithm to a solution of the constrained minimization problem (1.8).

**Theorem 3.1** Assume that  $\mathcal{B}_\tau(u) \equiv \mathcal{B}_\tau$  is a constant independent of  $u$  and  $\|g - g_\Omega\|_\tau^\tau \geq \nu_\tau |\Omega|$ . Then the pAPS-algorithm generates a sequence  $(\alpha_n)_n$  such that  $\lim_{n \rightarrow \infty} \alpha_n = \bar{\alpha} > 0$  with  $\mathcal{H}_\tau(u_{\bar{\alpha}}; g) = \lim_{n \rightarrow \infty} \mathcal{H}_\tau(u_{\alpha_n}; g) = \mathcal{B}_\tau$  and  $u_{\alpha_n} \rightarrow u_{\bar{\alpha}} \in \arg \min_{u \in X} \mathcal{J}_\tau(u, \bar{\alpha})$  for  $n \rightarrow \infty$ .

*Proof* Let us start with assuming that  $\mathcal{H}_\tau(u_{\alpha_0}; g) \leq \mathcal{B}_\tau$ . By induction, we show that  $\alpha_n \leq \alpha_{n+1}$  and  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau$ . In particular, if  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{B}_\tau$ , then  $\alpha_{n+1} = \left(\frac{\mathcal{B}_\tau}{\mathcal{H}_\tau(u_{\alpha_n}; g)}\right)^p \alpha_n > \alpha_n$ , where  $p > 0$  such that  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau$ ; cf. pAPS-algorithm. Then by Lemma 2.1 it follows that

$$\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau.$$

Note that there exists an  $\alpha^* > 0$  with  $\mathcal{H}_\tau(u_{\alpha^*}; g) = \mathcal{B}_\tau$ , see Theorem 2.2, such that for any  $\alpha \geq \alpha^*$ ,  $\mathcal{H}_\tau(u_\alpha; g) \geq \mathcal{B}_\tau$ ; cf. Lemma 2.1. If  $\alpha_n \geq \alpha^*$ , then  $\mathcal{H}_\tau(u_{\alpha_n}; g) \geq \mathcal{B}_\tau$ . Hence  $\mathcal{H}_\tau(u_{\alpha_n}; g) = \mathcal{B}_\tau$  and  $\alpha_{n+1} = \alpha_n$ . Thus, we deduce that the sequences  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  and  $(\alpha_n)_n$  are non-decreasing and bounded. Consequently, there exists an  $\bar{\alpha}$  such that  $\lim_{n \rightarrow \infty} \alpha_n = \bar{\alpha}$  with  $\mathcal{H}_\tau(u_{\bar{\alpha}}; g) = \mathcal{B}_\tau$ . Let  $\bar{\mathcal{H}} = \lim_{n \rightarrow \infty} \mathcal{H}_\tau(u_{\alpha_n}; g)$ . Then  $\bar{\mathcal{H}} = \mathcal{H}_\tau(u_{\bar{\alpha}}; g) = \mathcal{B}_\tau$ . By the optimality of  $u_{\alpha_n}$ , we have that  $0 \in \partial \mathcal{J}_\tau(u_{\alpha_n}, \alpha_n) = \partial \mathcal{H}_\tau(u_{\alpha_n}; g) + \alpha_n \partial \int_\Omega |Du_{\alpha_n}|$ ; see [39, Prop. 5.6 + Eq. (5.21), p. 26]. Consequently, there exist  $v_{\alpha_n} \in \partial \int_\Omega |Du_{\alpha_n}|$  such that  $-\alpha_n v_{\alpha_n} \in \partial \mathcal{H}_\tau(u_{\alpha_n}; g)$  with  $\lim_{n \rightarrow \infty} v_{\alpha_n} = v_{\bar{\alpha}}$ . By [79, Theorem. 24.4, p. 233], we obtain that  $-\bar{\alpha} v_{\bar{\alpha}} \in \partial \mathcal{H}_\tau(u_{\bar{\alpha}}; g)$  with  $v_{\bar{\alpha}} \in \partial \int_\Omega |Du_{\bar{\alpha}}|$  and hence  $0 \in \partial \mathcal{J}_\tau(u_{\bar{\alpha}}, \bar{\alpha})$  for  $n \rightarrow \infty$ .

If  $\mathcal{H}_\tau(u_{\alpha_0}; g) > \mathcal{B}_\tau$ , then as above we can show by induction that  $\alpha_n \geq \alpha_{n+1}$  and  $\mathcal{H}_\tau(u_{\alpha_n}; g) \geq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \geq \mathcal{B}_\tau$ . Thus, we deduce that  $(\mathcal{H}_\tau(u_{\alpha_n}; g))_n$  and  $(\alpha_n)_n$  are nonincreasing and bounded. Note that there exists an  $\alpha^* > 0$  with  $\mathcal{H}_\tau(u_{\alpha^*}; g) = \mathcal{B}_\tau$  such that for any  $\alpha \leq \alpha^*$ ,  $\mathcal{H}_\tau(u_\alpha; g) \leq \mathcal{B}_\tau$ . Hence if  $\alpha_n \leq \alpha^*$ , then  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{B}_\tau$ . This implies that  $\mathcal{H}_\tau(u_{\alpha_n}; g) = \mathcal{B}_\tau$  and  $\alpha_{n+1} = \alpha_n$ . The rest of the proof is identical to the above.  $\square$

*Remark 3.1* The adaptive choice of the value  $p$  in the pAPS-algorithm is fundamental for proving convergence in Theorem 3.1. In particular, the value  $p$  is chosen in dependency of  $\alpha$ , i.e., actually  $p = p(\alpha)$ , such that in the case of a constant  $\mathcal{B}_\tau$  the function  $\alpha \mapsto \frac{\mathcal{H}_\tau(u_\alpha; g)^{p(\alpha)}}{\alpha}$  is nonincreasing; cf. Fig. 5a.

### 3.2.2 The Non $p$ -Adaptive Case

A special case of the pAPS-algorithm accrues when the value  $p$  is not adapted in each iteration but set fixed. For the case  $p = 1$  (fixed), we obtain the following automated parameter selection algorithm.

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

- 1) Compute  $u_{\alpha_n} \in \arg \min_{u \in BV(\Omega)} \mathcal{J}_\tau(u, \alpha_n)$
- 2) Update  $\alpha_{n+1} := \frac{\mathcal{B}_\tau(u_{\alpha_n})}{\mathcal{H}_\tau(u_{\alpha_n}; g)} \alpha_n$  if  $\mathcal{H}_\tau(u_{\alpha_n}; g) > 0$  and continue with step 3). Otherwise increase  $\alpha_n$ , e.g.,  $\alpha_n := 10\alpha_n$ , and go to step 1).
- 3) Stop or set  $n := n + 1$  and return to step 1).

Even in this case, although under certain assumptions, we can immediately argue the convergence of this algorithm.

**Theorem 3.2** For  $\alpha > 0$ , let  $u_\alpha$  be a minimizer of  $\mathcal{J}_\tau(u, \alpha)$ . Assume that  $\mathcal{B}_\tau(u) \equiv \mathcal{B}_\tau$  is a constant independent of  $u$ , the function  $\alpha \mapsto \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  is nonincreasing, and  $\|g - g_\Omega\|_\tau^\tau \geq \nu_\tau |\Omega|$ . Then the APS-algorithm generates a sequence  $(\alpha_n)_n \subset \mathbb{R}^+$  such that  $\lim_{n \rightarrow \infty} \alpha_n = \bar{\alpha} > 0$ ,  $\lim_{n \rightarrow \infty} \mathcal{H}_\tau(u_{\alpha_n}; g) = \mathcal{B}_\tau$ , and  $u_{\alpha_n}$  converges to  $u_{\bar{\alpha}} \in \arg \min_{u \in BV(\Omega)} \mathcal{J}(u, \bar{\alpha})$  for  $n \rightarrow \infty$ .

*Proof* We only consider the case when  $\mathcal{H}_\tau(u_{\alpha_0}; g) \leq \mathcal{B}_\tau$  by noting that the case  $\mathcal{H}_\tau(u_{\alpha_0}; g) > \mathcal{B}_\tau$  can be shown analogous. By induction, we can show that  $\alpha_n \leq \alpha_{n+1}$  and  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau$ . More precisely, if  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{B}_\tau$ , then  $\alpha_{n+1} = \frac{\mathcal{B}_\tau}{\mathcal{H}_\tau(u_{\alpha_n}; g)} \alpha_n \geq \alpha_n$  and by Lemma 2.1 it follows that  $\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g)$ . Moreover, by the assumption that  $\alpha \mapsto \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  is nonincreasing, we obtain  $\mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \frac{\alpha_{n+1}}{\alpha_n} \mathcal{H}_\tau(u_{\alpha_n}; g) = \mathcal{B}_\tau$ . That is,

$$\mathcal{H}_\tau(u_{\alpha_n}; g) \leq \mathcal{H}_\tau(u_{\alpha_{n+1}}; g) \leq \mathcal{B}_\tau.$$

The rest of the proof is analogous to the one of Theorem 3.1.  $\square$

Nothing is known about the convergence of the APS-algorithm, if  $\mathcal{B}_\tau(\cdot)$  indeed depends on  $u$  and  $\mathcal{B}_\tau(u_{\alpha_n})$  is used instead of a fixed constant. In particular, in our numerics for some examples, in particular for the application of removing random-valued impulse noise with  $r = 0.05$ , we even observe that starting from a certain iteration the



sequence  $(\alpha_n)_n$  oscillates between two states, see Fig. 10c. This behavior can be attributed to the fact that, for example, if  $H_\tau(u_{\alpha_n}) \leq \mathcal{B}_\tau(u_{\alpha_n})$ , then it is not guaranteed that also  $H_\tau(u_{\alpha_{n+1}}) \leq \mathcal{B}_\tau(u_{\alpha_{n+1}})$ , which is essential for the convergence. The second assumption in the previous theorem, i.e., the nonincrease of the function  $\alpha \mapsto \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$ , can be slightly loosened, since for the convergence of the APS-algorithm it is enough to demand the nonincrease starting from a certain iteration  $\tilde{n} \geq 0$ . That is, if there exists a region  $U \subset \mathbb{R}^+$  where  $\alpha \mapsto \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  is nonincreasing and  $(\alpha_n)_{n \geq \tilde{n}} \subset U$ , then the algorithm converges; see Fig. 5. Analytically, this can be easily shown via Theorem 3.2 by just considering  $\alpha_{\tilde{n}}$  as the initial value of the algorithm. If  $\tau = 2$ , similar to the CPS-algorithm, we are able to show the following monotonicity property.

**Proposition 3.3** *If there exists a constant  $c > 0$  such that  $\|T^*(Tu - g)\|_{L^2(\Omega)} = c\|Tu - g\|_{L^2(\Omega)}$  for all  $u \in L^2(\Omega)$ , then the function  $\alpha \mapsto \frac{\sqrt{\mathcal{H}_2(u_\alpha; g)}}{\alpha}$  is nonincreasing, where  $u_\alpha$  is a minimizer of  $\mathcal{J}_2(u, \alpha)$ .*

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

$$\begin{aligned} \tilde{\mathcal{S}}(u, a) &:= \mathcal{J}_2(u, \alpha) + \frac{\delta}{2}\|u - a\|_{L^2(\Omega)}^2 - \frac{1}{2}\|T(u - a)\|_{L^2(\Omega)}^2 \\ &= \delta\|u - z(a)\|_{L^2(\Omega)}^2 + 2\alpha \int_{\Omega} |Du| + \psi(a, g, T), \end{aligned}$$

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

$$u_{\alpha,0} \in X, \quad u_{\alpha,k+1} = \arg \min_u \tilde{\mathcal{S}}(u, u_{\alpha,k}), \quad k \geq 0 \tag{3.4}$$

generates a sequence  $(u_{\alpha,k})_k$  which converges weakly for  $k \rightarrow \infty$  to a minimizer  $u_\alpha$  of  $\mathcal{J}_2(u, \alpha)$ ; see for example [32]. The unique minimizer  $u_{\alpha,k+1}$  is given by  $u_{\alpha,k+1} = (I - P_{\frac{\alpha}{\delta}K})(z(u_{\alpha,k}))$ , where  $K$  is the closure of the set

$$\{\operatorname{div} \xi : \xi \in C_c^1(\Omega, \mathbb{R}^2), |\xi(x)| \leq 1 \forall x \in \Omega\}$$

and  $P_K(u) := \arg \min_{v \in K} \|u - v\|_{L^2(\Omega)}$ ; see [18]. Then for  $k \rightarrow \infty$ , let us define

$$\tilde{f}(\alpha) := \left\| P_{\frac{\alpha}{\delta}K}(z(u_\alpha)) \right\|_{L^2(\Omega)} = \left\| \frac{1}{\delta}T^*(Tu_\alpha - g) \right\|_{L^2(\Omega)}.$$

Since  $\|T^*(Tu - g)\|_{L^2(\Omega)} = c\|Tu - g\|_{L^2(\Omega)}$ , it follows that  $\tilde{f}(\alpha) = \frac{c}{\delta}\sqrt{2\mathcal{H}_2(u_\alpha; g)}$ . The assertion follows by applying [18, Lemma 4.1], which is extendable to infinite dimensions, to  $\tilde{f}$  and by noting that the nonincrease of  $\alpha \mapsto \frac{\tilde{f}(\alpha)}{\alpha}$  implies the nonincrease of  $\alpha \mapsto \frac{\sqrt{\mathcal{H}_2(u_\alpha; g)}}{\alpha}$ .  $\square$

We remark that for convolution type of operators the assumption of Proposition 3.3 does not hold in general. However, there exist several operators  $T$ , relevant in image processing, with the property  $\|T^*(Tu - g)\|_{L^2(\Omega)} = \|Tu - g\|_{L^2(\Omega)}$ . Such operators include  $T = I$  for image denoising,  $T = 1_D$  for image inpainting, where  $1_D$  denotes the characteristic function of the domain  $D \subset \Omega$ , 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 [17] or in wavelet inpainting [25], respectively. For all such operators, the function  $\alpha \mapsto \frac{\sqrt{\mathcal{H}_2(u_\alpha; g)}}{\alpha}$  is nonincreasing and hence by setting  $p = \frac{1}{2}$  fixed in the pAPS-algorithm or changing the update of  $\alpha$  in the APS-algorithm to

$$\alpha_{n+1} := \sqrt{\frac{\mathcal{B}_2}{\mathcal{H}_2(u_{\alpha_n}; g)}}\alpha_n,$$

where  $\mathcal{B}_2$  is a fixed constant, chosen according to (1.6), we obtain in these situations a convergent algorithm.

We emphasize once more that in general the nonincrease of the function  $\alpha \mapsto \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  is not guaranteed. Nevertheless, there exists always a constant  $p \geq 0$  such that  $\alpha \mapsto \frac{(\mathcal{H}_\tau(u_\alpha; g))^p}{\alpha}$  is indeed nonincreasing. For example,  $p = \frac{1}{2}$  for operators  $T$  with the property  $\|T^*(Tu - g)\|_{L^2(\Omega)} = \|Tu - g\|_{L^2(\Omega)}$ ; cf. Proposition 3.3. In particular, one easily checks the following result.

**Proposition 3.4** *Let  $0 < \alpha \leq \beta$ , and  $u_\alpha$  and  $u_\beta$  minimizers of  $\mathcal{J}_\tau(\cdot, \alpha)$  and  $\mathcal{J}_\tau(\cdot, \beta)$ , respectively, for  $\tau = 1, 2$ . Then  $\frac{(\mathcal{H}_\tau(u_\beta; g))^p}{\beta} \leq \frac{(\mathcal{H}_\tau(u_\alpha; g))^p}{\alpha}$  if and only if  $p \leq \frac{\ln \beta - \ln \alpha}{\ln \mathcal{H}_\tau(u_\beta; g) - \ln \mathcal{H}_\tau(u_\alpha; g)}$ .*

### 4 Locally Constrained TV Problem

In order to enhance image details, while preserving homogeneous regions, we formulate, as in [37, 58], a locally constrained optimization problem. That is, instead of considering (1.6) we formulate

$$\min_{u \in BV(\Omega)} \int_{\Omega} |Du| \quad \text{s.t.} \quad \int_{\Omega} w(x, y)|Tu - g|^\tau(y)dy \leq \nu_\tau \tag{4.1}$$

for almost every  $x \in \Omega$ , where  $w$  is a normalized filter, i.e.,  $w \in L^\infty(\Omega \times \Omega)$ , and  $w \geq 0$  on  $\Omega \times \Omega$  with

$$\begin{aligned} \int_{\Omega} \int_{\Omega} w(x, y)dydx &= 1 \\ \text{and} & \\ \int_{\Omega} \int_{\Omega} w(x, y)|\phi(y)|^\tau dydx &\geq \epsilon \|\phi\|_{L^\tau(\Omega)}^\tau \end{aligned} \tag{4.2}$$

for all  $\phi \in L^\tau(\Omega)$  and for some  $\epsilon > 0$  independent of  $\phi$ ; cf. [37,58].

### 4.1 Local Filtering

In practice, for  $w$  we may use the mean filter together with a windowing technique; see for example [37,58]. In order to explain the main idea, we continue in a discrete setting. Let  $\Omega^h$  be a discrete image domain containing  $N_1 \times N_2$  pixels,  $N_1, N_2 \in \mathbb{N}$ , and by  $|\Omega^h| = N_1 N_2$  we denote the size of the discrete image (number of pixels). We approximate functions  $u$  by discrete functions, denoted by  $u^h$ . The considered function spaces are  $X = \mathbb{R}^{N_1 \times N_2}$  and  $Y = X \times X$ . In what follows for all  $u^h \in X$ , we use the following norms:

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

for  $1 \leq \tau < +\infty$ . Moreover, we denote by  $u_\Omega^h$  the average value of  $u^h \in X$ , i.e.,  $u_\Omega^h := \frac{1}{|\Omega^h|} \sum_{x \in \Omega^h} u^h(x)$ . The discrete gradient  $\nabla^h : X \rightarrow Y$  and the discrete divergence  $\text{div}^h : Y \rightarrow X$  are defined in a standard way by forward and backward differences such that  $\text{div}^h := -(\nabla^h)^*$ ; see for example [18,22,55,63]. With the above notations and definitions, the discretization of the general function in (1.9) is given by

$$J_\tau(u^h, \alpha) := H_\tau(u^h) + R_\alpha(u^h), \tag{4.3}$$

where  $H_\tau(u^h) = \frac{1}{\tau} \|T^h u^h - g^h\|_\tau^\tau$ ,  $\tau \in \{1, 2\}$ ,  $T^h : X \rightarrow X$  is a bounded linear operator,  $\alpha \in (\mathbb{R}^+)^{N_1 \times N_2}$ , and

$$R_\alpha(u^h) := \sum_{x \in \Omega^h} \alpha(x) |\nabla^h u^h(x)|_{l^2} \tag{4.4}$$

with  $|y|_{l^2} = \sqrt{y_1^2 + y_2^2}$  for every  $y = (y_1, y_2) \in \mathbb{R}^2$ . In the sequel, if  $\alpha$  is a scalar or  $\alpha \equiv 1$  in (4.4), we write instead of  $R_\alpha$  or  $R_1$  just  $\alpha R$  or  $R$ , respectively, i.e.,

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

is the discrete total variation of  $u$  in  $\Omega^h$ , and we write  $\bar{E}_\tau$  instead of  $E_\tau$  to indicate that  $\alpha$  is constant. Introducing some step-size  $h$ , then for  $h \rightarrow 0$  (i.e. the number of pixels  $N_1 N_2$  goes to infinity), one can show, similar as for the case  $\alpha \equiv 1$ , that  $R_\alpha$   $\Gamma$ -converges to  $\int_\Omega \alpha |Du|$ ; see [12,62]. We turn now to the locally constrained minimization problem, which is given in the discrete setting as

$$\min_{u^h \in X} R(u^h) \quad \text{s.t.} \quad S_{i,j}^\tau(u^h) \leq \frac{\nu_\tau}{\tau} \quad \text{for all } x_{i,j} \in \Omega^h. \tag{4.5}$$

Here  $\nu_\tau$  is a fixed constant and

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

denotes the local residual at  $x_{i,j} \in \Omega^h$  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 [37,54,58] for  $\mathcal{I}_{i,j}$  the set

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

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

$$\begin{aligned} \tilde{\Omega}_{i,j}^\omega &= \left\{ x_{s+i,t+j} : \max \left\{ 1 - (i, j), -\frac{\omega-1}{2} \right\} \leq (s, t) \right. \\ &\quad \left. \leq \min \left\{ \frac{\omega-1}{2}, (N_1 - i, N_2 - j) \right\} \right\}. \end{aligned}$$

Hence  $\tilde{\Omega}_{i,j}^\omega \subset \Omega^h$  for all  $x_{i,j} \in \Omega^h$ . Before we analyze the difference between  $\Omega_{i,j}^\omega$  and  $\tilde{\Omega}_{i,j}^\omega$  with respect to the constrained minimization problem (4.5), we note that since  $T^h$  does not annihilate constant functions, the existence of a solution of (4.5) is guaranteed; see [37, Theorem 2][58, Theorem 2].

In the following, we set  $B_\tau := \frac{\nu_\tau}{\tau} |\Omega^h|$ .

- Proposition 4.1** (i) If  $u^h$  is a solution of (4.5) with  $\mathcal{I}_{i,j} = \Omega_{i,j}^\omega$ , then  $H_\tau(u^h) < B_\tau$ .  
(ii) If  $u$  is a solution of (4.5) with  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}^\omega$ , then  $H_\tau(u^h) \leq B_\tau$ .

*Proof* (i) Since  $u^h$  is a solution of (4.5) and  $\Omega_{i,j}^\omega$  is a set of pixels in a  $\omega$ -by- $\omega$  window, we have

$$\begin{aligned} B_\tau &\geq \sum_{i,j} S_{i,j}^\tau(u^h) \\ &= \sum_{i,j} \frac{1}{\tau \omega^2} \sum_{x_{s,t} \in \Omega_{i,j}^\omega} |g^h(x_{s,t}) - T^h u^h(x_{s,t})|^\tau \\ &> \frac{1}{\tau} \sum_{i,j} |g^h(x_{i,j}) - T^h u^h(x_{i,j})|^\tau = H_\tau(u^h). \end{aligned}$$

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

(ii) For a minimizer  $u^h$  of (4.5), we obtain

$$\begin{aligned} B_\tau &\geq \sum_{i,j} S_{i,j}^\tau(u^h) \\ &= \sum_{i,j} \frac{1}{\tau M_{i,j}} \sum_{x_{s,t} \in \Omega_{i,j}^\omega} |g^h(x_{s,t}) - T^h u^h(x_{s,t})|^\tau \\ &= \frac{1}{\tau} \sum_{i,j} |g^h(x_{i,j}) - T^h u^h(x_{i,j})|^\tau = H_\tau(u^h), \end{aligned}$$

which concludes the proof. □

Note that if  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}^\omega$ , then by Proposition 4.1 a minimizer of (4.5) also satisfies the constraint of the problem

$$\min_{u^h \in X} R(u^h) \quad \text{s.t.} \quad H_\tau(u^h) \leq B_\tau \tag{4.6}$$

(discrete version of (2.1)) but is in general of course not a solution of (4.6).

**Proposition 4.2** *Let  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}^\omega$ ,  $u_s^h$  be a minimizer of (4.6), and  $u_l^h$  be a minimizer of (4.5). Then  $R(u_s^h) \leq R(u_l^h)$ .*

*Proof* Assume that  $R(u_s^h) > R(u_l^h)$ . Since  $u_s^h$  is a solution of (4.6), it satisfies the constraint  $H_\tau(u_s^h) \leq B_\tau$ . By Proposition 4.1, we also have  $H_\tau(u_l^h) \leq B_\tau$ . Since  $R(u_s^h) > R(u_l^h)$ ,  $u_s^h$  is not the solution of (4.6) which is a contradiction. Hence,  $R(u_s^h) \leq R(u_l^h)$ . □

*Remark 4.1* Proposition 4.1 and its consequence are not special properties of the discrete setting. Let the filter  $w$  in (4.1) be such that the inequality in (4.2) becomes an equality with  $\epsilon = 1/|\Omega|$ , as it is the case in Proposition 4.1(ii). Then a solution  $u_l$  of the locally constrained minimization problem (4.1) satisfies

$$\mathcal{H}_\tau(u_l; g) \leq \frac{\nu_\tau}{\tau} |\Omega| \quad \text{and} \quad \int_\Omega |Du_l| \geq \int_\Omega |Du_s|,$$

where  $u_s$  is a solution of (2.1).

From Proposition 4.2 and Remark 4.1, we conclude, since  $R(u_s^h) \leq R(u_l^h)$  and  $\int_\Omega |Du_s| \leq \int_\Omega |Du_l|$ , that  $u_s^h$  and  $u_s$  are smoother than  $u_l^h$  and  $u_l$ , 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. Since noise can be interpreted as fine details, which we actually want to eliminate, this could also mean that noise is possibly left in the image.

### 4.2 Locally Adaptive Total Variation Algorithm

Whenever  $\nu_\tau$  depends on  $\hat{u}$ , problem (4.5) results in a quite nonlinear problem. Instead of considering nonlinear constraints, we choose as in Sect. 3 a reference image  $\tilde{u}$  and

compute an approximate  $\nu_\tau = \nu_\tau(\tilde{u})$ . Note that in our discrete setting for salt-and-pepper noise we have now

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

and for random-valued impulse noise we have

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

In our below proposed locally adaptive algorithms, we choose as a reference image the current approximation (see LATV- and pLATV-algorithm below), as also done in the pAPS- and APS-algorithm above. Then we are seeking for a solution  $u^h$  such that  $S_{i,j}^\tau(u^h)$  is close to  $\frac{\nu_\tau}{\tau}$ . We note that for large  $\alpha > 0$  the minimization of (4.3) yields an over-smoothed restoration  $u_\alpha^h$  and the residual contains details, i.e., we expect  $H_\tau(u_\alpha^h) > B_\tau$ . Hence, if  $S_{i,j}^\tau(u_\alpha^h) > \frac{\nu_\tau}{\tau}$ , we suppose that this is due to image details contained in the local residual image. In this situation, we intend to decrease  $\alpha$  in the local regions  $\mathcal{I}_{i,j}$ . In particular, we define, similar as in [37,58], the local quantity  $f_{i,j}^\omega$  by

$$f_{i,j}^\omega := \begin{cases} S_{i,j}^\tau(u_\alpha^h) & \text{if } S_{i,j}^\tau(u_\alpha^h) > \frac{\nu_\tau}{\tau}, \\ \frac{\nu_\tau}{\tau} & \text{otherwise.} \end{cases}$$

Note that  $\frac{\nu_\tau}{\tau f_{i,j}^\omega} \leq 1$  for all  $i, j$  and hence we set

$$\alpha(x_{i,j}) := \frac{1}{M_{i,j}} \sum_{x_{s,t} \in \mathcal{I}_{i,j}} \left( \frac{\nu_\tau}{\tau f_{s,t}^\omega} \right)^p \alpha(x_{s,t}). \tag{4.7}$$

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

$$f_{i,j}^\omega := \begin{cases} S_{i,j}^\tau(u_\alpha^h) & \text{if } S_{i,j}^\tau(u_\alpha^h) < \frac{\nu_\tau}{\tau}, \\ \frac{\nu_\tau}{\tau} & \text{otherwise,} \end{cases}$$

and setting  $\alpha$  as in (4.7). Notice that now  $\frac{\nu_\tau}{\tau f_{i,j}^\omega} \geq 1$ . 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  $u_{\alpha_n}^h \in \arg \min_{u^h \in X} J_\tau(u^h, \alpha_n)$
- 2) (a) If  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$ , then set

$$f_{i,j}^\omega := \max \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\}$$

- (b) If  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$ , then set

$$f_{i,j}^\omega := \max \left\{ \min \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\}, \varepsilon \right\}$$

- 3) Update

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

- 4) 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  $f_{i,j}^\omega > 0$ , since it may happen that  $S_{i,j}^\tau(u_{\alpha_n}^h) = 0$ .

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

The LATV-algorithm has the following monotonicity properties with respect to  $(\alpha_n)_n$ .

**Proposition 4.3** Assume  $\mathcal{I}_{i,j} = \Omega_{i,j}^\omega$  and let  $\varepsilon > 0$  be sufficiently small. If  $\alpha_0 > 0$  such that  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$ , 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 4.1, we obtain

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

Note that  $v_\tau(\cdot)$  is bounded from below, see Sect. 3.1. Consequently, there exists an  $\varepsilon > 0$  such that  $\frac{v_\tau(u^h)}{\tau} \geq \varepsilon$  for any

$u^h$ . Then, since  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$  we have by the LATV-algorithm that

$$f_{i,j}^\omega := \max \left\{ \min \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\}, \varepsilon \right\} \leq \frac{v_\tau(u_{\alpha_n}^h)}{\tau}$$

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

**Proposition 4.4** Let  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}^\omega$  and  $\varepsilon > 0$  be sufficiently small.

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

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

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

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

*Proof* (i) By the same argument as in the proof of Proposition 4.1 and since  $f_{i,j}^\omega := \max \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\} \geq \frac{v_\tau(u_{\alpha_n}^h)}{\tau}$ , we obtain

$$\begin{aligned} \sum_{i,j} \alpha_{n+1}(x_{i,j}) &= \sum_{i,j} \left( \frac{v_\tau(u_{\alpha_n}^h)^p}{\tau^p M_{i,j}} \sum_{x_{s,t} \in \tilde{\Omega}_{i,j}^\omega} \frac{\alpha_n(x_{s,t})}{(f_{s,t}^\omega)^p} \right) \\ &= \sum_{i,j} \left( \left( \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right)^p \frac{\alpha_n(x_{i,j})}{(f_{i,j}^\omega)^p} \right) \\ &\leq \sum_{i,j} \alpha_n(x_{i,j}). \end{aligned}$$

- (ii) Since  $v_\tau(\cdot)$  is bounded from below, see Sect. 3.1, there exists an  $\varepsilon > 0$  such that  $\frac{v_\tau(u^h)}{\tau} \geq \varepsilon$  for any  $u^h$ . Hence

$$\begin{aligned} f_{i,j}^\omega &:= \max \left\{ \min \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\}, \varepsilon \right\} \\ &\leq \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \end{aligned}$$

and by the same arguments as above we get

$$\begin{aligned} \sum_{i,j} \alpha_{n+1}(x_{i,j}) &= \sum_{i,j} \left( \left( \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right)^p \frac{\alpha_n(x_{i,j})}{(f_{i,j}^\omega)^p} \right) \\ &\geq \sum_{i,j} \alpha_n(x_{i,j}). \end{aligned}$$

$\square$

In contrast to the pAPS-algorithm in the LATV-algorithm, the power  $p > 0$  is not changed during the iterations and should be chosen sufficiently small, e.g., we set  $p = \frac{1}{2}$  in our experiments. Note that small  $p$  only allows small changes of  $\alpha$  in each iteration. In this way, the algorithm is able to generate a function  $\alpha^*$  such that  $H_\tau(u_{\alpha^*}^h)$  is very close to  $\frac{v_\tau(u_{\alpha^*}^h)}{\tau}$ . On the contrary, small  $p$  has the drawback that the number of iterations till termination is kept large. Since the parameter  $p$  has to be chosen manually, the LATV-algorithm, at least in the spirit, seems to be similar to Uzawa’s method, where also a parameter has to be chosen. The proper choice of such a parameter might be complicated and hence we are desiring for an algorithm where we do not have to tune parameters manually. Because of this and motivated by the pAPS-algorithm, we propose the following  $p$  adaptive algorithm:

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

0) Compute  $u_{\alpha_n} \in \arg \min_{u^h \in X} J_\tau(u^h, \alpha_n)$

1) (a) If  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$ , then set

$$f_{i,j}^\omega := \max \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\}$$

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

$$f_{i,j}^\omega := \max \left\{ \min \left\{ S_{i,j}^\tau(u_{\alpha_n}^h), \frac{v_\tau(u_{\alpha_n}^h)}{\tau} \right\}, \varepsilon \right\}$$

2) Update

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

3) Compute  $u_{\alpha_{n+1}}^h \in \arg \min_{u^h \in X} J_\tau(u^h, \alpha_{n+1})$

4) (a) if  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$

(i) if  $H_\tau(u_{\alpha_{n+1}}^h) \leq B_\tau(u_{\alpha_{n+1}}^h)$ , go to step 5)

(ii) if  $H_\tau(u_{\alpha_{n+1}}^h) > B_\tau(u_{\alpha_{n+1}}^h)$ , decrease  $p$ , e.g., set  $p = p/10$ , and go to step 2)

(b) if  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$

(i) if  $H_\tau(u_{\alpha_{n+1}}^h) \geq B_\tau(u_{\alpha_{n+1}}^h)$ , go to step 5)

(ii) if  $H_\tau(u_{\alpha_{n+1}}^h) < B_\tau(u_{\alpha_{n+1}}^h)$ , 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  $|H_\tau(u_{\alpha_n}^h) - B_\tau(u_{\alpha_n}^h)| \leq 10^{-6}$  and  $H_\tau(u_{\alpha_n}^h) \leq B_\tau(u_{\alpha_n}^h)$ . Additionally, we stop iterating when  $p$  is less than machine precision, since then anyway no progress is to be expected. Due to the adaptive choice of  $p$ , we obtain a monotonic behavior of the sequence  $(\alpha_n)_n$ .

**Proposition 4.5** *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  $\alpha_0$  such that  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$  and monotonically increasing for  $\alpha_0$  such that  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$ .*

*Proof* For  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$ , we can show by induction that by the pLATV-algorithm  $f_{i,j}^\omega \geq \frac{v_\tau(u_{\alpha_n}^h)}{\tau}$  and hence  $1 \geq \frac{v_\tau(u_{\alpha_n}^h)}{\tau f_{i,j}^\omega}$  for all  $n$ . Then by the definition of  $\alpha_{n+1}$  it follows

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

By similar arguments, we obtain for  $\alpha_0$  with  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$  that  $\alpha_{n+1}(x_{i,j}) \geq \alpha_n(x_{i,j})$  for all  $x_{i,j} \in \Omega$ .  $\square$

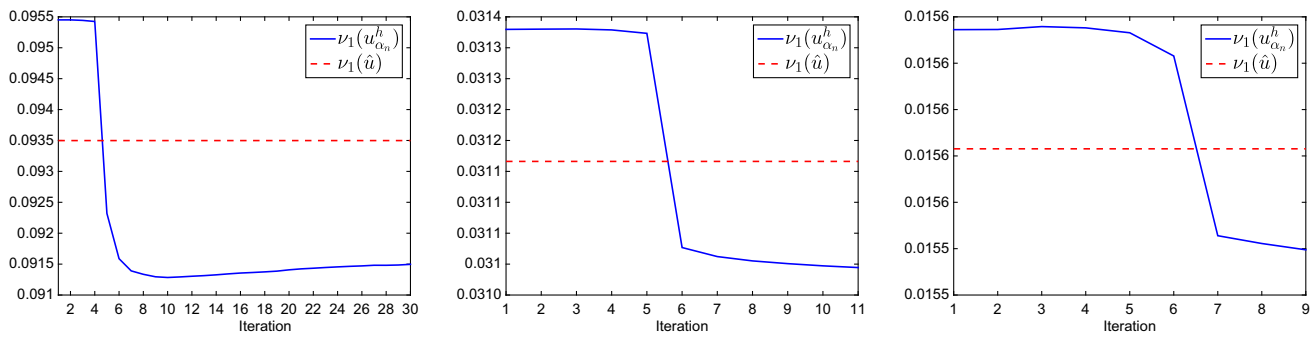
We are aware of the fact that using  $u_{\alpha_n}^h$  as a reference image in the LATV- and pLATV-algorithm to compute  $v_\tau$  may commit errors. However, we recall that for Gaussian noise we set  $v_2 = \sigma^2$  and for salt-and-pepper noise with  $r_1 = r_2$  we have  $v_1 = r_1$ . In these cases,  $v_\tau$  does not depend on the original image and hence we do not commit any error by computing  $v_\tau$ . For random-valued impulse noise-corrupted images, the situation is different and  $v_1$  indeed depends on the true image. In this situation, errors may be committed when  $u_{\alpha_n}^h$  is used as a reference image for calculating  $v_\tau$ ; see Figs. 2 and 3. Hence, in order to improve the proposed algorithm, for such cases for future research it might be of interest to find the optimal reference image to obtain a good approximation of the real value  $v_\tau$ .

In contrast to the SA-TV algorithm presented in [37,58], 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  $\alpha_0$  can be chosen arbitrarily positive. However, in the case  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$  we cannot guarantee in general that the solution  $u_\alpha$  obtained by the pLATV-algorithm fulfills  $H_\tau(u_\alpha^h) \leq B_\tau(u_\alpha^h)$ , not even if  $B_\tau(\cdot)$  is constant, due to the stopping criterion with respect to the power  $p$ . On the contrary, if  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$ , then the pLATV-algorithm generates a sequence  $(u_{\alpha_n}^h)_n$  such that  $H_\tau(u_{\alpha_n}^h) \leq B_\tau(u_{\alpha_n}^h)$  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  $H_\tau(u_{\alpha_0}^h) \leq B_\tau(u_{\alpha_0}^h)$ , which may be realized by the following simple automated procedure:

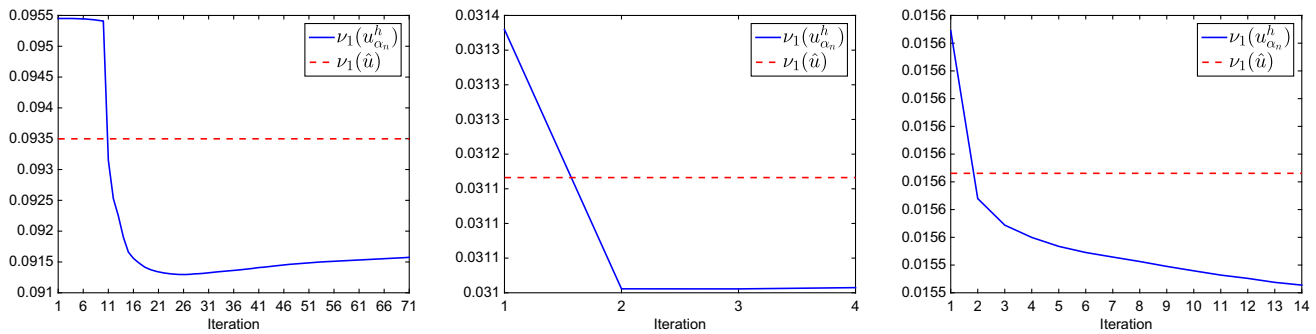
**Algorithm 1:** Input:  $\alpha_0 > 0$  (arbitrary);

1) Compute  $u_{\alpha_0}^h \in \arg \min_{u^h \in X} J_\tau(u^h, \alpha_0)$ .

2) If  $H_\tau(u_{\alpha_0}^h) > B_\tau(u_{\alpha_0}^h)$  decrease  $\alpha_0$  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  $\alpha_0$ .



**Fig. 2** Progress of  $\nu_\tau(u_{\alpha_n}^h)$  of the LATV-algorithm with  $p = \frac{1}{8}$  and  $\alpha_0 = 10^{-2}$  for removing random-valued impulse noise with  $r = 0.3$  (left),  $r = 0.1$  (middle), and  $r = 0.05$  (right) from the cameraman-image (cf. Fig. 4b)



**Fig. 3** Progress of  $\nu_\tau(u_{\alpha_n}^h)$  of the pLATV-algorithm with  $p_0 = \frac{1}{2}$  and  $\alpha_0 = 10^{-2}$  for removing random-valued impulse noise with  $r = 0.3$  (left),  $r = 0.1$  (middle), and  $r = 0.05$  (right) from the cameraman-image (cf. Fig. 4b)

### 5 Total Variation Minimization

In this section, we are concerned with developing numerical methods for computing a minimizer of the discrete multiscale  $L^\tau$ -TV model, i.e.,

$$\min_{u^h \in X} \{J_\tau(u^h, \alpha) := H_\tau(u^h) + R_\alpha(u^h)\}. \tag{5.1}$$

#### 5.1 $L^2$ -TV Minimization

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

$$\min_{u^h \in X} \frac{1}{2} \|T^h u^h - g^h\|_2^2 + R_\alpha(u^h), \tag{5.2}$$

and present solution methods, first for the case  $T^h = I$  and then for general linear bounded operators  $T^h$ .

##### 5.1.1 An Algorithm for Image Denoising

If  $T^h = I$ , then (5.2) becomes an image denoising problem, i.e., the minimization problem

$$\min_{u^h \in X} \|u^h - g^h\|_2^2 + 2R_\alpha(u^h). \tag{5.3}$$

For solving this problem, we use the algorithm of Chambolle and Pock [22], which leads to the following iterative scheme:

**Chambolle–Pock algorithm:** Initialize  $\tau, \sigma > 0, \theta \in [0, 1], (\mathbf{p}_0^h, u_0^h) \in Y \times X$ , set  $\bar{u}_0^h = u_0^h$ , and set  $n = 0$ .

1. Compute
 
$$\mathbf{p}_{n+1}^h(x) = \frac{\mathbf{p}_n^h(x) + \sigma \nabla^h \bar{u}_n^h(x)}{\max \left\{ \frac{1}{\alpha(x)} |\mathbf{p}_n^h(x) + \sigma \nabla^h \bar{u}_n^h(x)|, 1 \right\}},$$
 for all  $x \in \Omega^h$ .
2. Compute  $u_{n+1}^h = \frac{u_n^h + \tau \operatorname{div}^h \mathbf{p}_{n+1}^h + \tau g^h}{1 + \tau}$ .
3. Set  $\bar{u}_{n+1}^h = u_{n+1}^h + \theta(u_{n+1}^h - u_n^h)$ .
4. Stop or set  $n := n + 1$  and return to step 1).

In our numerical experiments, we choose  $\theta = 1$ . In particular, in [22] it is shown that for  $\theta = 1$  and  $\tau\sigma \|\nabla^h\|^2 < 1$  the algorithm converges.

##### 5.1.2 An Algorithm for Linear Bounded Operators

Assume that  $T^h$  is a linear bounded operator from  $X$  to  $X$ , different to the identity  $I$ . Then instead of minimizing (5.2) directly, we introduce the surrogate functional

$$\begin{aligned} \mathcal{S}(u^h, a^h) &:= \frac{1}{2} \|T^h u^h - g^h\|_2^2 + R_\alpha(u^h) + \frac{\delta}{2} \|u^h - a^h\|_2^2 \\ &\quad - \frac{1}{2} \|T^h(u^h - a^h)\|_2^2 \tag{5.4} \\ &= \frac{\delta}{2} \|u^h - z(a^h)\|_2^2 + R_\alpha(u^h) + \psi(a^h, g^h, T^h) \end{aligned}$$

with  $a^h, u^h \in X, z(a^h) = a^h - \frac{1}{\delta}(T^h)^*(T^h a^h - g^h), \psi$  a function independent of  $u^h$ , and where we assume  $\delta > \|T^h\|^2$ ; see [31,32]. Note that

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

and hence to obtain a minimizer amounts to solve a minimization problem of the type (5.3) and can be solved as described in Sect. 5.1.1. Then an approximate solution of (5.2) can be computed by the following iterative algorithm: Choose  $u_0^h \in X$  and iterate for  $n \geq 0$

$$u_{n+1}^h = \arg \min_{u^h \in X} \mathcal{S}(u^h, u_n^h). \tag{5.5}$$

For scalar  $\alpha$ , it is shown in [28,31,32] that this iterative procedure generates a sequence  $(u_n^h)_n$  which converges to a minimizer of (5.2). This convergence property can be easily extended to our nonscalar case yielding the following result.

**Theorem 5.1** For  $\alpha : \Omega \rightarrow \mathbb{R}^+$  the scheme in (5.5) generates a sequence  $(u_n^h)_n$ , which converges to a solution of (5.2) for any initial choice of  $u_0^h \in X$ .

*Proof* A proof can be accomplished analogous to [32].  $\square$

### 5.2 An Algorithm for $L^1$ -TV Minimization

The computation of a minimizer of

$$\min_{u^h \in X} \|T^h u^h - g^h\|_1 + R_\alpha(u^h), \tag{5.6}$$

is due to the nonsmooth  $\ell^1$ -term in general more complicated than obtaining a solution of the  $L^2$ -TV model. Here we suggest to employ a trick, proposed in [5] for  $L^1$ -TV minimization problems with a scalar regularization parameter, to solve (5.6) in two steps. In particular, we substitute the argument of the  $\ell^1$ -norm by a new variable  $v$ , penalize the functional by an  $L^2$ -term, which should keep the difference between  $v$  and  $Tu - g$  small, and minimize with respect to  $v$  and  $u$ . That is, we replace the original minimization (5.6) by

$$\min_{v^h, u^h \in X} \|v^h\|_1 + \frac{1}{2\gamma} \|T^h u^h - g^h - v^h\|_2^2 + R_\alpha(u^h), \tag{5.7}$$

where  $\gamma > 0$  is small, so that we have  $g^h \approx T^h u^h - v^h$ . Actually, it can be shown that (5.7) converges to (5.6) as  $\gamma \rightarrow 0$ . In our experiments, we actually choose  $\gamma = 10^{-2}$ . This leads to the following alternating algorithm.

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

1) Compute

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

2) Compute

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

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

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

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

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

$$\arg \min_{u^h \in X} \frac{1}{2} \|T^h u^h - g^h - v_{n+1}^h\|_2^2 + R_{\gamma\alpha}(u^h) \tag{5.8}$$

and hence is of the type (5.2). Thus, an approximate solution of (5.8) can be computed as described above; see Sect. 5.1.

**Theorem 5.2** The sequence  $(u_n^h, v_n^h)_n$  generated by the  $L^1$ -TV $_\alpha$  algorithm converges to a minimizer of (5.7).

*Proof* The statement can be shown analogous to [5].  $\square$

### 5.3 A Primal–Dual Method for $L^1$ -TV Minimization

For solving (1.6) with  $\tau = 1$ , we suggest, alternatively to the above method, to use the primal–dual method of [58] adapted to our setting, where a Huber regularization of the gradient of  $u$  is considered; see [58] for more details. Denoting by  $\bar{u}$  a corresponding solution of the primal problem and  $\bar{\mathbf{p}}$  the solution of the associated dual problem, the optimality conditions due to the Fenchel theorem [39] are given by

$$\begin{aligned}
 -\operatorname{div} \bar{\mathbf{p}}(x) &= -\kappa \Delta \bar{u}(x) + \frac{1}{\beta + \mu} T^*(T\bar{u}(x) - g(x)) \\
 &\quad + \frac{\mu}{\beta + \mu} T^* \frac{T\bar{u}(x) - g(x)}{\max\{\beta, |T\bar{u}(x) - g(x)|\}} \\
 -\bar{\mathbf{p}}(x) &= \frac{1}{\gamma} \nabla \bar{u}(x) \quad \text{if } |\bar{\mathbf{p}}(x)|_2 < \alpha(x) \\
 -\bar{\mathbf{p}}(x) &= \alpha(x) \frac{\nabla \bar{u}(x)}{|\nabla \bar{u}(x)|_2} \quad \text{if } |\bar{\mathbf{p}}(x)|_2 = \alpha(x),
 \end{aligned}$$

for all  $x \in \Omega$ , where  $\kappa, \beta, \mu$ , and  $\gamma$  are fixed positive constants. The latter two conditions can be summarized to  $-\bar{\mathbf{p}}(x) = \frac{\alpha(x)\nabla \bar{u}(x)}{\max\{\gamma\alpha(x), |\nabla \bar{u}(x)|_2\}}$ . 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:

$$\begin{aligned}
 0 &= -\max\{\beta, |T\bar{u}(x) - g(x)|\} \bar{v} + T\bar{u}(x) - g(x) \\
 0 &= \operatorname{div} \bar{\mathbf{q}}(x) + \kappa \Delta \bar{u}(x) - \frac{1}{\beta + \mu} T^*(T\bar{u}(x) - g(x)) \\
 &\quad - \frac{\mu}{\beta + \mu} T^* \bar{v}(x) \tag{5.9} \\
 0 &= \max\{\gamma\alpha(x), |\nabla \bar{u}(x)|_2\} \bar{\mathbf{q}}(x) - \alpha(x) \nabla \bar{u}(x)
 \end{aligned}$$

for all  $x \in \Omega$ . This system can be solved efficiently by a semi-smooth Newton algorithm; see Appendix for a description of the method and for the choice of the parameters  $\kappa, \beta, \mu$ , and  $\gamma$ .

Note that different algorithms presented in the literature can also be adjusted to the case of a locally varying regularization parameter, such as [18, 21, 68]. However, it is not the scope of this paper to compare different algorithms in order to detect the most efficient one, although this is an interesting research topic in its own right.

## 6 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  $\alpha_0$ . 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) [90], 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

$$\text{MAE} = \|u - \hat{u}\|_{L^1(\Omega)},$$

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 the MAE becomes, the better the reconstruction results are.

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

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

$$\frac{|\mathcal{H}_\tau(u_{\alpha_n}; g) - \mathcal{B}_\tau(u_{\alpha_n})|}{\mathcal{B}_\tau(u_{\alpha_n})} \leq \epsilon_B = 10^{-5}$$

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 be expected. In fact, if our algorithm converges at least linearly, i.e., if there exists an  $\epsilon_\alpha \in (0, 1)$  and an  $m > 0$  such that for all  $n \geq m$  we have  $\|\alpha_{n+1} - \alpha_\infty\| < \epsilon_\alpha \|\alpha_n - \alpha_\infty\|$ , the second stopping criterion at least ensures that the distance between our obtained result  $\alpha$  and  $\alpha_\infty$  is  $\|\alpha - \alpha_\infty\| \leq \frac{\epsilon_\alpha \epsilon_\alpha}{1 - \epsilon_\alpha}$ .

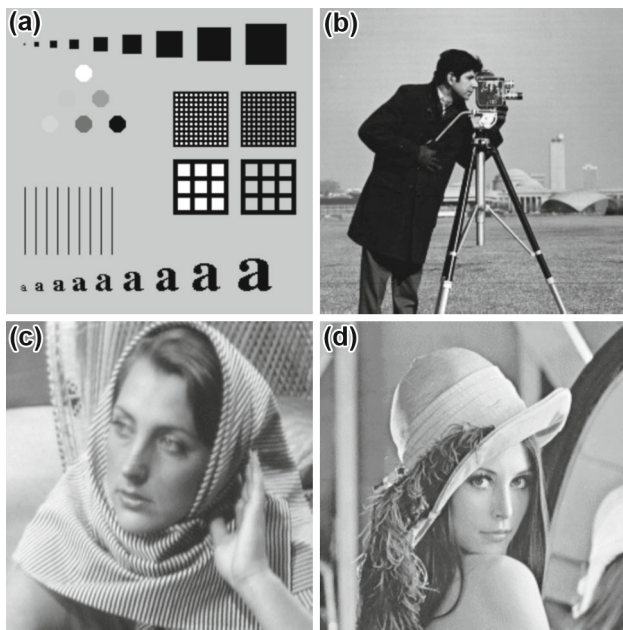
### 6.1 Automatic Scalar Parameter Selection

For automatically selecting the scalar parameter  $\alpha$  in (1.6), we presented in Sect. 3 the APS- and pAPS-algorithm. Here we compare their performance for image denoising and image deblurring (Fig. 4).

#### 6.1.1 Gaussian Noise Removal

For recovering images corrupted by Gaussian noise with mean zero and standard deviation  $\sigma$ , we minimize the functional in (1.6) by setting  $\tau = 2$  and  $T = I$ . Then  $\mathcal{B}_2 = \frac{\sigma^2}{2} |\Omega|$  is a constant independent of  $u$ . The automatic selection of a suitable regularization parameter  $\alpha$  is here performed by the CPS-, APS-, and pAPS-algorithm, where the contained minimization problem is solved by the method presented in Sect. 5.1.1. We recall that by [18, Theorem 4] and Theorem 3.1 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 (1.8). In particular, in the pAPS-algorithm the value  $p$  is chosen in dependency of  $\alpha$ , i.e.,  $p = p(\alpha)$ , such that  $\alpha \rightarrow \frac{(\mathcal{H}_\tau(u_\alpha; g))^{p(\alpha)}}{\alpha}$  is nonincreasing; see Fig. 5a. This property is fundamental for obtaining convergence of this algorithm; see Theorem 3.1. For the APS-algorithm, such a monotonic behavior is not guaranteed and hence we cannot ensure its convergence. Nevertheless, if the APS-algorithm generates  $\alpha$ 's such that the function  $\alpha \rightarrow \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  is nonincreasing, then it indeed converges to the desired solution; see



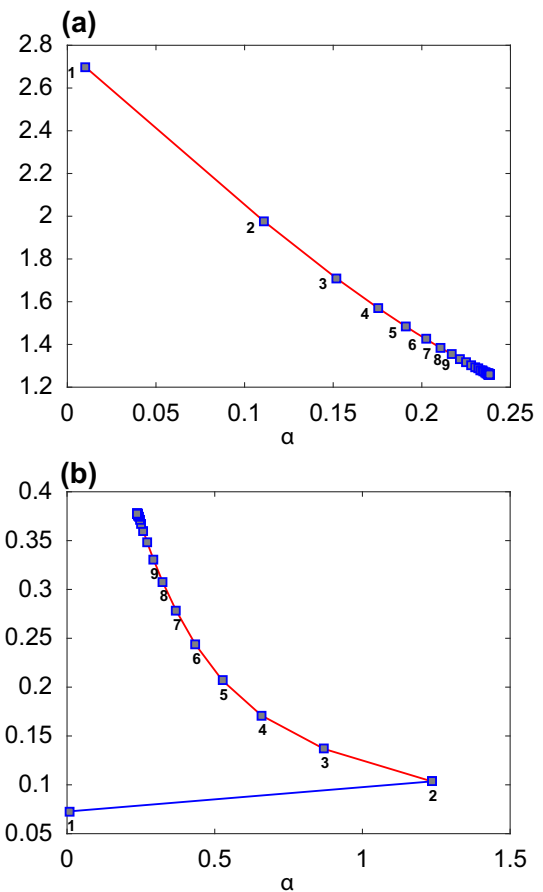


**Fig. 4** Original images. **a** Phantom, **b** cameraman, **c** barbara, and **d** lena

**Theorem 3.2.** Unfortunately, the nonincrease of the function  $\alpha \rightarrow \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  does not hold always, see Fig. 5b.

For performance comparison, here we consider the phantom-image of size  $256 \times 256$  pixels, see Fig. 4a, corrupted only by Gaussian white noise with  $\sigma = 0.3$ . In the pAPS-algorithm we set  $p_0 = 32$ . The behavior of the sequence  $(\alpha_n)_n$  for different initial  $\alpha_0$ , i.e.,  $\alpha_0 \in \{1, 10^{-1}, 10^{-2}\}$  is depicted in Fig. 6. We observe that all three methods for arbitrary  $\alpha_0$  converge to the same regularization parameter  $\alpha$  and hence generate results with the same PSNR and MSSIM (i.e., PSNR= 19.84 and MSSIM= 0.7989). Hence, in these experiments, despite the lack of theoretical convergence, also the APS-algorithm seems to converge to the desired solutions. We observe the same behavior for different  $\sigma$  as well.

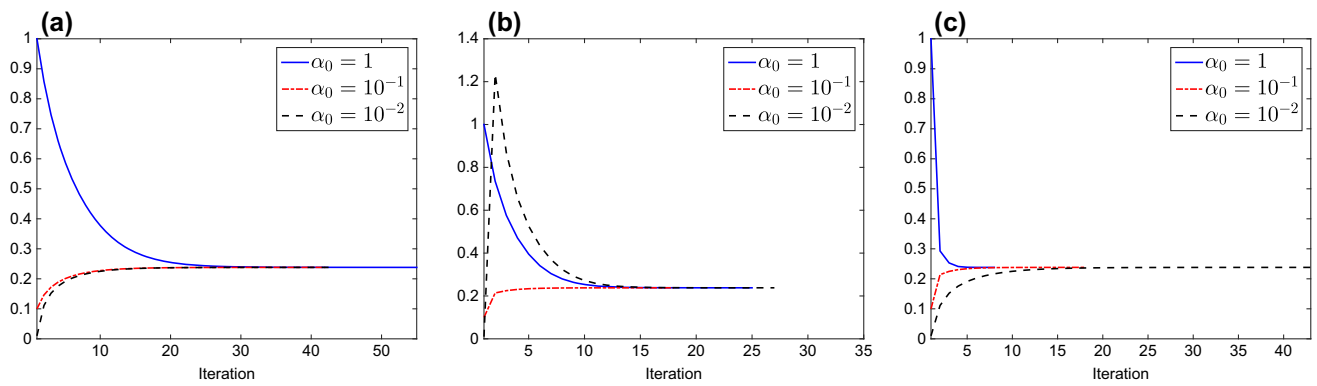
Looking at the number of iterations needed till termination, we observe from Table 1 that the APS-algorithm always needs significantly less iterations than the CPS-algorithm till termination. This is attributed to the different updates of  $\alpha$ . Recall that for a fixed  $\alpha_n$  in the CPS-algorithm we set  $\alpha_{n+1}^{CPS} := \sqrt{\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)}}\alpha_n$ , while in the APS-algorithm the update is performed as  $\alpha_{n+1}^{APS} := \frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)}\alpha_n$ . Note that  $\sqrt{\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)}} \leq \frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)}$ , if  $\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)} \geq 1$  and  $\sqrt{\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)}} \geq \frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)}$ , if  $\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)} \leq 1$ . Hence, we obtain  $\alpha_n \leq \alpha_{n+1}^{CPS} \leq \alpha_{n+1}^{APS}$  if  $\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)} \geq 1$  and  $\alpha_n \geq \alpha_{n+1}^{CPS} \geq \alpha_{n+1}^{APS}$  if  $\frac{v_2|\Omega|}{\tau\mathcal{H}_2(u_{\alpha_n}; g)} \leq 1$ . That is, in the APS-algorithm  $\alpha$  changes more significantly in each iteration than in the CPS-algorithm, which leads to a faster conver-



**Fig. 5** Denoising of the phantom-image corrupted with Gaussian white noise with  $\sigma = 0.03$ . **a** Plot of the function  $\alpha \rightarrow \frac{\mathcal{H}_\tau(u_\alpha; g)^p}{\alpha}$  of the pAPS-algorithm with  $\alpha_0 = 10^{-2}$ . **b** Plot of the function  $\alpha \rightarrow \frac{\mathcal{H}_\tau(u_\alpha; g)}{\alpha}$  of the APS-algorithm with  $\alpha_0 = 10^{-2}$ . The desired monotone behavior is observed after the second iteration (red part of the curve)

gence with respect to the number of iterations. Nevertheless, exactly this behavior allows the function  $\alpha \rightarrow \frac{\mathcal{H}_2(u_\alpha; g)}{\alpha}$  to increase which is responsible for that the convergence of the APS-algorithm is not guaranteed in general. However, in our experiments we observed that the function  $\alpha \rightarrow \frac{\mathcal{H}_2(u_\alpha; g)}{\alpha}$  only increases in the first iterations, but does not increase (actually even decreases) afterwards; see Fig. 5b. This is actually enough to guarantee convergence, as discussed in Sect. 3, 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 nonincrease holds, we get convergence of the algorithm.

The pAPS-algorithm is designed to ensure the nonincrease of the function  $\alpha \rightarrow \frac{(\mathcal{H}_\tau(u_\alpha; g))^{p(\alpha)}}{\alpha}$  by choosing  $p(\alpha)$  in each iteration accordingly, which is done by the algorithm automatically. If  $p(\alpha) = p = 1/2$  in each iteration, then the pAPS-algorithm becomes the CPS-algorithm, as it happens sometimes in practice (indicated by the same number of iterations in Table 1). Since the CPS-algorithm converges [18],



**Fig. 6** Denoising of the phantom-image corrupted with Gaussian white noise with  $\sigma = 0.03$ . **a** CPS-algorithm, **b** APS-algorithm, and **c** pPAS-algorithm

**Table 1** Number of iterations needed for the reconstruction of the phantom-image corrupted by Gaussian white noise with different standard deviations  $\sigma$

$\alpha_0$	$\sigma = 0.3$			$\sigma = 0.1$			$\sigma = 0.05$			$\sigma = 0.01$		
	CPS	APS	pAPS	CPS	APS	pAPS	CPS	APS	pAPS	CPS	APS	pAPS
1	55	25	8	47	21	21	46	20	20	47	17	47
$10^{-1}$	42	18	18	38	17	6	44	20	20	46	18	46
$10^{-2}$	43	27	43	40	20	40	40	18	40	39	17	6
$10^{-3}$	43	31	43	41	22	41	41	19	41	41	20	41
$10^{-4}$	43	35	43	41	23	41	41	22	41	41	19	41

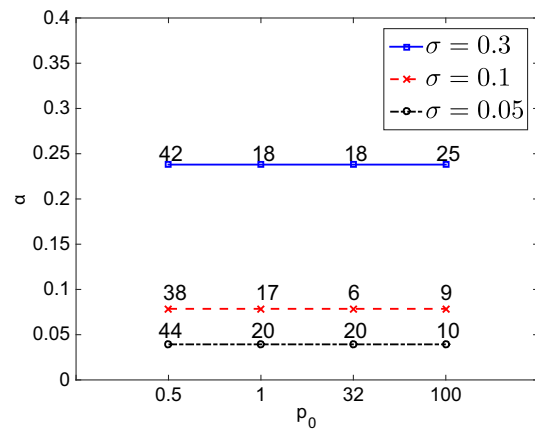
In the pAPS-algorithm we set  $p_0 = 32$

the pAPS-algorithm always yields  $p \geq 1/2$ . In particular, we observe that if the starting value  $\alpha_0$  is larger than the requested regularization parameter  $\alpha$ , less iterations till termination are needed than with the CPS-algorithm. On the contrary, if  $\alpha_0$  is smaller than the desired  $\alpha$ ,  $p = 1/2$  is chosen by the algorithm to ensure the monotonicity. The obtained result of the pAPS-algorithm is independent of the choice of  $p_0$  as visible from Fig. 7. In this plot, we also specify the number of iterations needed till termination. On the optimal choice of  $p_0$  with respect to the number of iterations, we conclude from Fig. 7 that  $p_0 = 32$  seems to do a good job, although the optimal value may depend on the noise level.

Similar behaviors as described above are also observed for denoising other and real images as well.

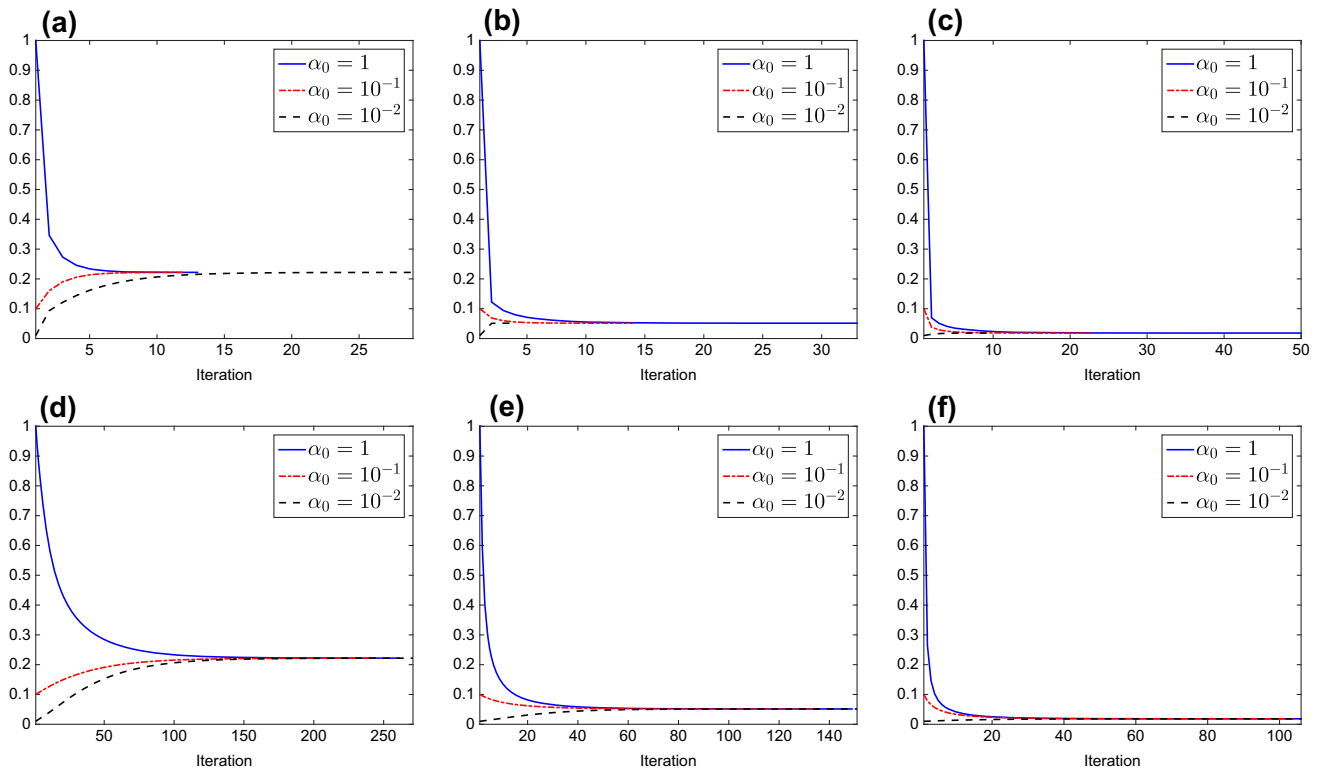
### 6.1.2 Image Deblurring

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 Fig. 4b, which is of size  $256 \times 256$  pixels, blur it by a Gaussian blurring kernel of size  $5 \times 5$  pixels and standard deviation 10, and additionally add some Gaussian white noise with variance  $\sigma^2$ . The minimization problem in the APS- and pAPS-algorithm is solved approximately



**Fig. 7** Regularization parameter  $\alpha$  obtained by the pAPS-algorithm with different  $p_0$  for denoising the phantom-image corrupted with Gaussian white noise for different  $\sigma$

by the algorithm in (5.5). In Fig. 8, the progress of  $\alpha_n$  for different  $\sigma$ 's, i.e.,  $\sigma \in \{0.3, 0.1, 0.05\}$ , and different  $\alpha_0$ 's, i.e.,  $\alpha_0 \in \{1, 10^{-1}, 10^{-2}\}$ , is presented. In these tests, both algorithms converge to the same regularization parameter and minimizer. From the figure, we observe that 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.



**Fig. 8** Reconstruction of the cameraman-image corrupted by Gaussian blurring kernel of size  $5 \times 5$  and standard deviation 10. In the pAPS-algorithm, we set  $p_0 = 32$ . **a**  $\sigma = 0.3$ ; pAPS-algorithm, **b**  $\sigma = 0.1$ ;

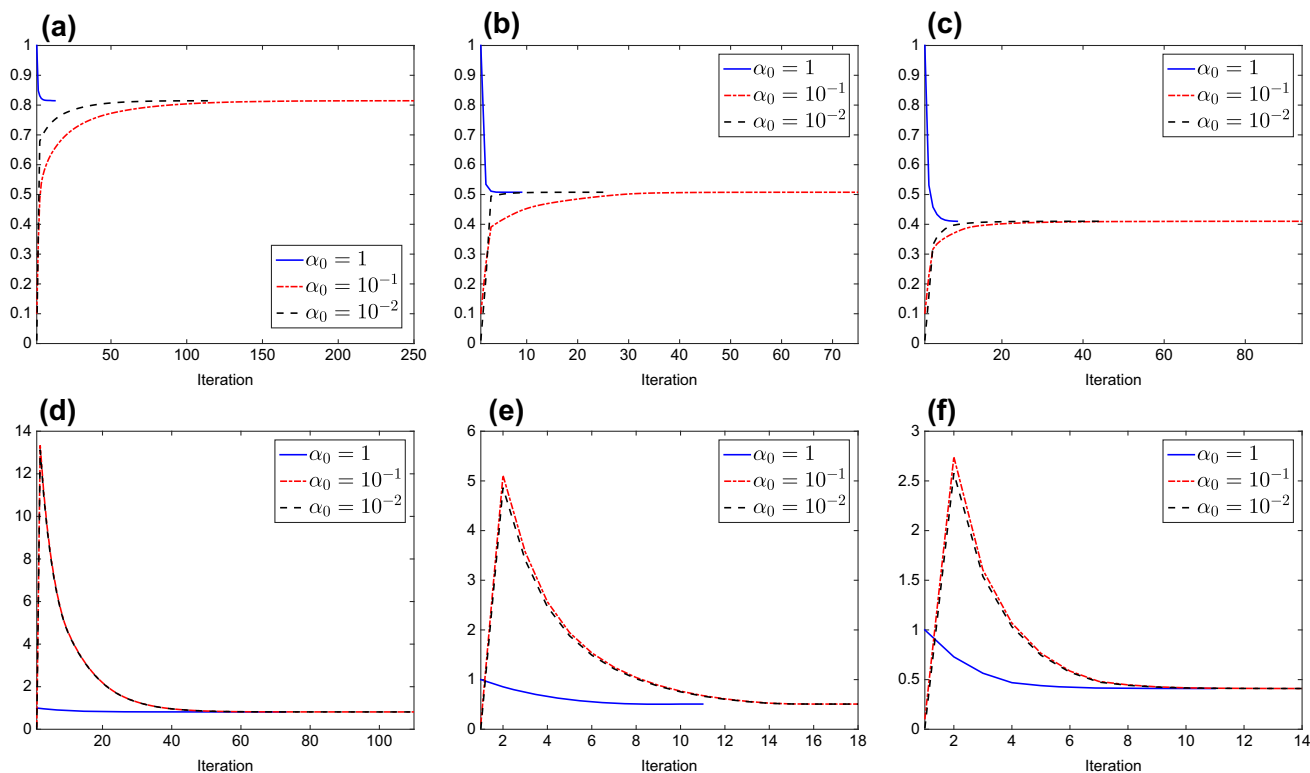
pAPS-algorithm, **c**  $\sigma = 0.05$ ; pAPS-algorithm, **d**  $\sigma = 0.3$ ; APS-algorithm, **e**  $\sigma = 0.1$ ; APS-algorithm, and **f**  $\sigma = 0.05$ ; APS-algorithm

### 6.1.3 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  $\alpha$  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 $_{\alpha}$ -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  $\alpha_0$ 's are depicted in Figs. 9 and 10. For the removal of salt-and-pepper noise, we observe from Fig. 9 similar behaviors of the APS- and pAPS-algorithm as above for removing Gaussian noise. In particular, both algorithms converge to the same regularization parameter. However, in many cases the APS-algorithm needs significantly less iterations than the pAPS-algorithm. These behaviors are also observed in Fig. 10 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 Fig. 10c.

### 6.2 Locally Adaptive Total Variation Minimization

In this section, various experiments are presented to evaluate the performance of the LATV- and pLATV-algorithm presented in Sect. 4. Their performance is compared with the proposed pAPS-algorithm as well as with the SA-TV-algorithm introduced in [37] for  $L^2$ -TV minimization and in [58] 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  $\lambda$  based on a local variance estimation. However, as pointed out in [37, 58], they only perform efficiently when the initial  $\lambda$  is chosen sufficiently small, as we will do in our numerics. On the contrary, for the LATV- and pLATV-algorithm any positive initial  $\alpha_0$  is sufficient. For the comparison, we consider four different images, shown in Fig. 4, which are all of size  $256 \times 256$  pixels. In all our experiments, for the SA-TV-algorithm we use  $\mathcal{I}_{i,j} = \Omega_{i,j}^{\omega}$ , see [37], and we set the window-size to  $11 \times 11$  pixels in the case of Gaussian noise and to  $21 \times 21$  pixels in case of impulse noise. For the LATV- and pLATV-algorithm, we use the window-size  $\omega = 11$ , if not otherwise specified, and choose  $p_0 = \frac{1}{2}$ .



**Fig. 9** Denoising of the cameraman-image corrupted by salt-and-pepper noise. In the pAPS-algorithm, we set  $p_0 = 32$ . **a**  $r_1 = r_2 = 0.3$ ; pAPS-algorithm, **b**  $r_1 = r_2 = 0.1$ ; pAPS-algorithm, **c**  $r_1 = r_2 = 0.05$ ;

pAPS-algorithm, **d**  $r_1 = r_2 = 0.3$ ; APS-algorithm, **e**  $r_1 = r_2 = 0.1$ ; APS-algorithm, and **f**  $r_1 = r_2 = 0.05$ ; APS-algorithm

### 6.3 Gaussian Noise Removal

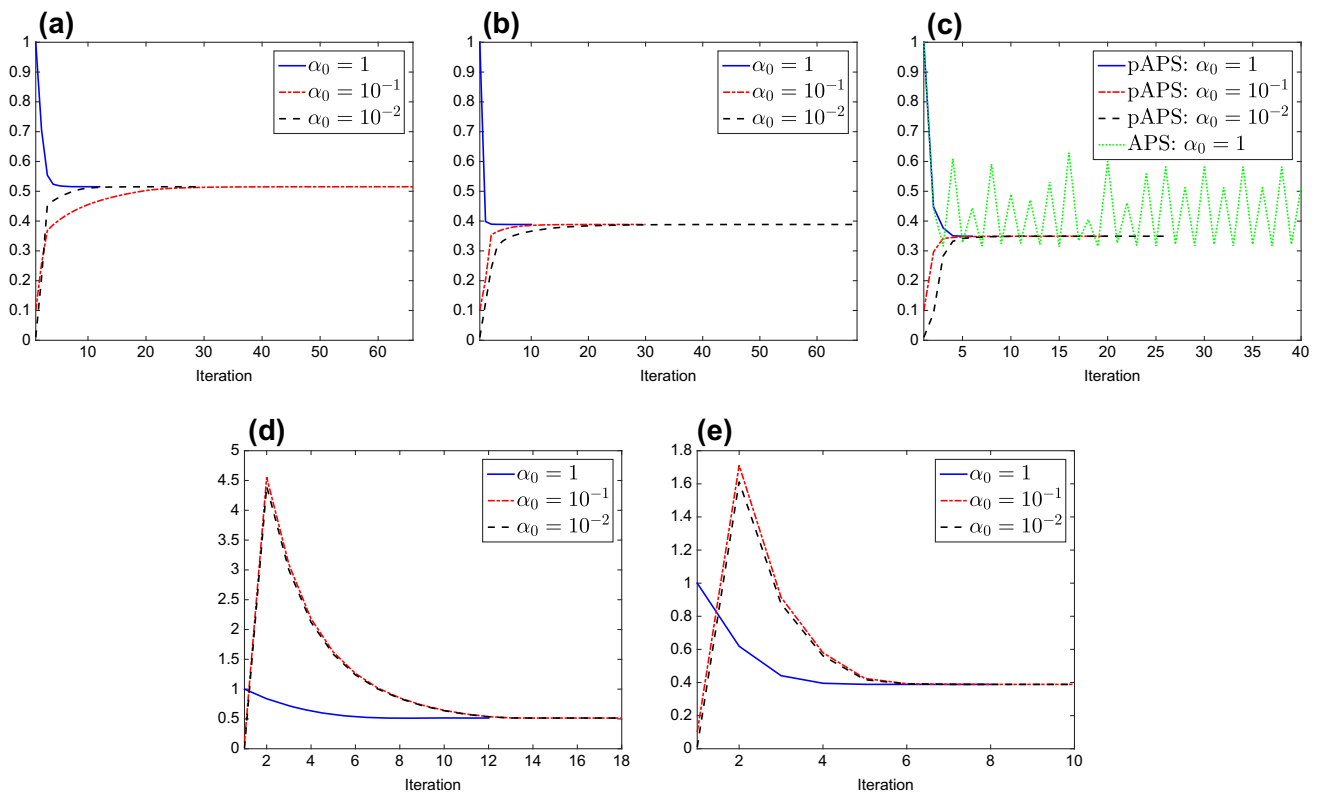
#### 6.3.1 Dependency on the Initial Regularization Parameter

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.,  $\lambda_0$  for the SA-TV-algorithm and  $\alpha_0$  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 parameter, where we set  $c_{\alpha_0} = \frac{1}{5}$ . The minimization problems contained in the LATV- and pLATV-algorithm are solved as described in Sect. 5.1.1. For comparison reasons, we define the values  $PSNR_{diff} := \max_{\alpha_0} PSNR(\alpha_0) - \min_{\alpha_0} PSNR(\alpha_0)$  and  $MSSIM_{diff} := \max_{\alpha_0} MSSIM(\alpha_0) - \min_{\alpha_0} MSSIM(\alpha_0)$  to measure the variation of the considered quality measures. Here  $PSNR(\alpha_0)$  and  $MSSIM(\alpha_0)$  are the PSNR and MSSIM-values of the reconstructions, respectively, which are obtained from the considered algorithms when the initial regularization parameter is set to  $\alpha_0$ . From Table 2, we observe that the pLATV-algorithm with and without Algorithm 1 is 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  $\alpha_0$ . As expected, the pLATV-algorithm with Algorithm 1 is even more stable with respect to  $\alpha_0$  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  $\alpha_0$  is sufficiently small, then the pLATV-algorithm with and without Algorithm 1 coincides; see Table 2 for  $\alpha_0 \in \{10^{-2}, 10^{-3}, 10^{-4}\}$ . Actually, in the rest of our experiments we choose  $\alpha_0$  always so small that Algorithm 1 returns the inputted  $\alpha_0$ .

#### 6.3.2 Dependency on the Local Window

In Table 3, we report on the performance tests of the pLATV-algorithm with respect to the chosen type of window, i.e.,  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}^\omega$  and  $\mathcal{I}_{i,j} = \Omega_{i,j}^\omega$ . 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 windows. Nevertheless, the boundaries



**Fig. 10** Denoising of the cameraman-image corrupted by random-valued impulse noise. In the pAPS-algorithm, we set  $p_0 = 32$ . **a**  $r = 0.3$ ; pAPS-algorithm, **b**  $r = 0.1$ ; pAPS-algorithm, **c**  $r = 0.05$ ; pAPS-algorithm, **d**  $r = 0.3$ ; APS-algorithm, and **e**  $r = 0.1$ ; APS-algorithm

**Table 2** 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  $\alpha_0$  and via the SA-TV-algorithm with different  $\lambda_0$ . In the LATV- and pLATV-algorithm, we use  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$  with window-size  $11 \times 11$  pixels in the interior and we set  $p_0 = \frac{1}{2}$

$\alpha_0/\lambda_0$	SA-TV		LATV		pLATV		pLATV with Algorithm 1	
	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
PSNR <sub>diff</sub>	0.39646		0.20257		0.044473		0.012704	
MSSIM <sub>diff</sub>	0.036322		0.0091312		0.0092963		0.00019843	

are treated differently, which leads to different theoretical results, but seems not to have a 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 4. Since for both types of windows nearly the same results are obtained, in the rest of our experiments we limit ourselves to always set  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}^\omega$  in the LATV- and pLATV-algorithm.

Next, we test the pLATV-algorithm for different values of the window-size varying from 3 to 15. Fig. 11 shows the PSNR and MSSIM of the restoration of the cameraman-image degraded by different types of noise (i.e., Gaussian

noise with  $\sigma = 0.3$  or  $\sigma = 0.1$ , salt-and-pepper noise with  $r_1 = r_2 = 0.3$  or  $r_1 = r_2 = 0.1$ , or random-valued impulse noise with  $r = 0.3$  or  $r = 0.1$ ), where the pLATV-algorithm with  $\alpha_0 = 10^{-2}$  and  $p_0 = 1/2$  is used. We observe that the PSNR and MSSIM are varying only slightly with respect to changing window-size. However, in the case of Gaussian noise elimination the PSNR and MSSIM increase very slightly with increasing window-size, while in the case of impulse noise contamination such a behavior cannot be observed. In Fig. 11, we also specify the number of iterations needed till termination of the algorithm. From this, we observe that a larger window-size results in most experiments in more iterations.

**Table 3** PSNR and MSSIM of the reconstruction of different images corrupted by Gaussian white noise with standard deviation  $\sigma$  via the pLATV-algorithm with  $\alpha_0 = 10^{-4}$

Image	$\sigma$	pLATV with $\mathcal{I} = \tilde{\Omega}$		pLATV with $\mathcal{I} = \Omega$	
		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

6.3.3 Homogeneous Noise

Now, we test the algorithms for different images corrupted by Gaussian noise with zero mean and different standard deviations  $\sigma$ , i.e.,  $\sigma \in \{0.3, 0.1, 0.05, 0.01\}$ . The initial regularization parameter  $\alpha_0$  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 4, 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  $\sigma$  is suf-

ficiently small. In Fig. 12, 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 oversmoothed. 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 4 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.

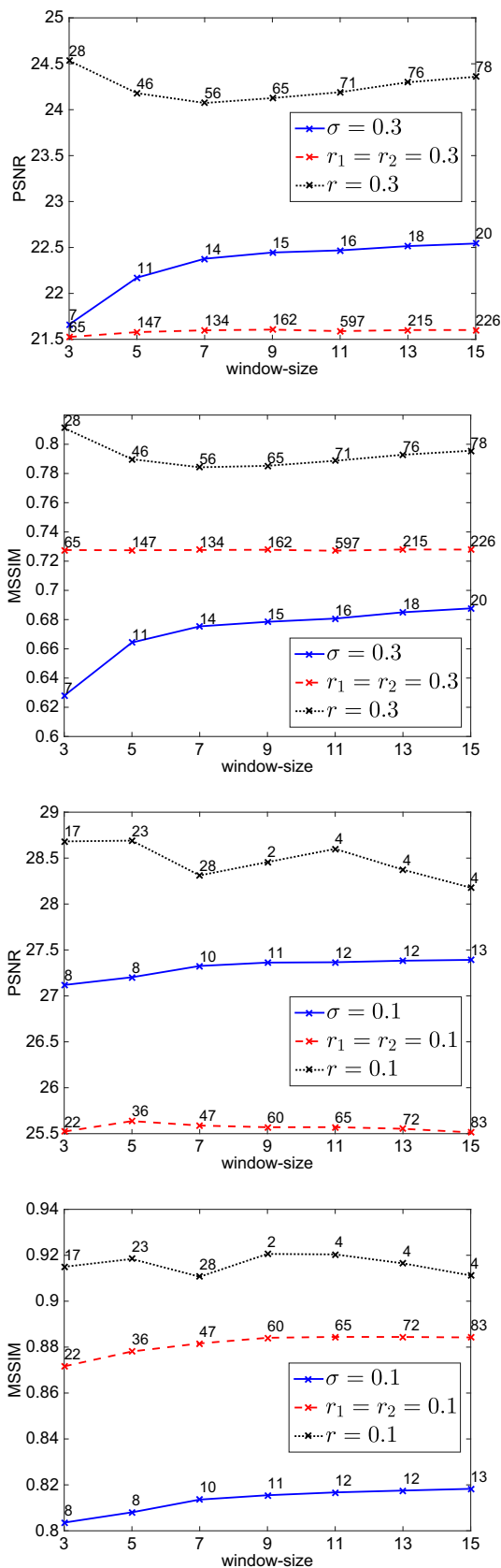
6.3.4 Nonhomogeneous Noise

For this experiment, we consider the cameraman-image degraded by Gaussian white noise with variance  $\sigma^2 = 0.0025$  in the whole domain  $\Omega$  except a rather small area (highlighted in red in Fig. 13a), denoted by  $\tilde{\Omega}$ , where the variance is 6 times larger, i.e.,  $\sigma^2 = 0.015$  in this part. Since the noise level is in this application not homogeneous, the pLATV-algorithm presented in Sect. 4 has to be adjusted to this situation accordingly. This can be done by making  $\nu_\tau$  (here  $\tau = 2$ ) locally dependent and we write  $\nu_\tau = \nu_\tau(\hat{u})(x)$  to stress the dependency on the true image  $\hat{u}$  and on the location  $x \in \Omega$  in the image. In particular, for our experiment we set  $\nu_2 = 0.015$  in  $\tilde{\Omega}$ , while  $\nu_2 = 0.0025$  in  $\Omega \setminus \tilde{\Omega}$ . Since  $\nu_\tau$  is now varying, we also have to adjust the definition of  $\mathcal{B}_\tau$  and  $B_\tau$  to

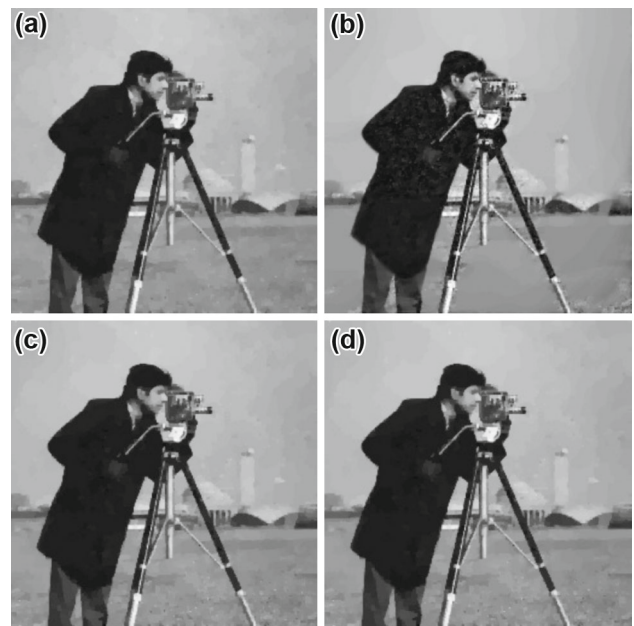
**Table 4** PSNR- and MSSIM-values of the reconstruction of different images corrupted by Gaussian white noise with standard deviation  $\sigma$  via pAPS-, LATV-, and pLATV-algorithm with  $\alpha_0 = 10^{-4}$  and SA-TV-algorithm with  $\lambda_0 = 10^{-4}$ . In the LATV- and pLATV-algorithm, we use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

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

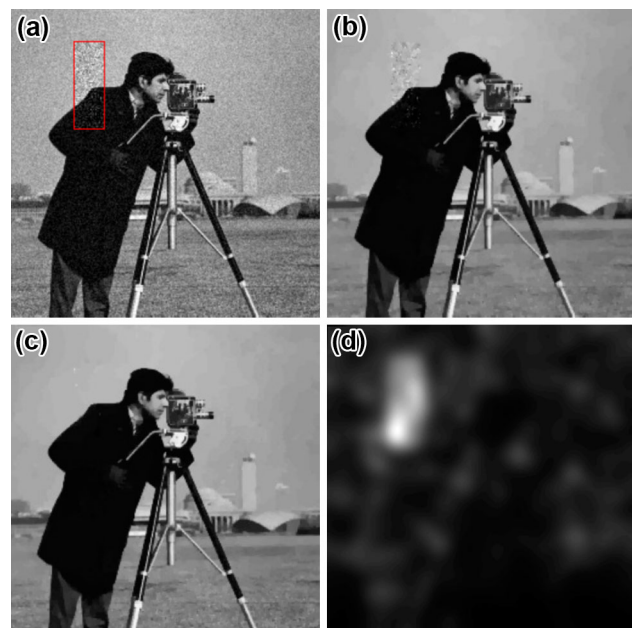
A bold values indicate the best value for the corresponding experiment



**Fig. 11** Restoration of the cameraman-image corrupted by different types of noise via the pLATV-method with different window-sizes



**Fig. 12** Reconstruction of the cameraman-image corrupted by Gaussian white noise with  $\sigma = 0.1$ . **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), and **d** pLATV (PSNR: 27.38; MSSIM: 0.8171)



**Fig. 13** Reconstruction of the cameraman-image corrupted by Gaussian white noise with  $\sigma^2 = 0.0025$  except in (a) the highlighted area where  $\sigma^2 = 0.015$

$$B_\tau(u) := \int_\Omega v_\tau(u)(x) \, dx \text{ and } B_\tau(u^h) := \sum_{x \in \Omega^h} v_\tau(u^h)(x),$$

respectively, for the continuous and discrete setting. Making these adaptations allows us to apply the pLATV-algorithm as well as the pAPS-algorithm to the application of removing nonuniform noise.

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

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

A bold values indicate the best value for the corresponding experiment

The reconstructions obtained by the pAPS-algorithm (with  $p_0 = 32$  and  $\alpha_0 = 10^{-2}$ ) and by the pLATV-algorithm (with  $p_0 = 1/2$  and  $\alpha_0 = 10^{-2}$ ) are shown in Figs. 13b and c, respectively. Due to the adaptive choice of  $\alpha$ , see Fig. 13d where light colors indicate a large value, the pLATV-algorithm is able to remove all the noise considerably, while the pAPS-algorithm returns a restoration, which still retains noise in  $\hat{\Omega}$ .

#### 6.4 Deblurring and Gaussian Noise Removal

The performance of the algorithms for restoring images corrupted by Gaussian blur with blurring kernel of size  $5 \times 5$  pixels and standard deviation 10 and additive Gaussian noise with standard deviation  $\sigma$  is reported in Table 5. Here we observe that the LATV- as well as the pLATV-algorithm outperforms the SA-TV-algorithm for nearly any example. This observation is also clearly visible in Fig. 14, 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  $\sigma$  the LATV-algorithm has always the best PSNR.

#### 6.5 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 [58], where a semismooth Newton method is used to generate an estimate of the minimizer of (1.10). For the sake of a fair comparison, an approximate solution of the minimization problem in



**Fig. 14** Reconstruction of the cameraman-image corrupted by Gaussian white noise with  $\sigma = 0.1$  and Gaussian blur. **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), and **d** pLATV (PSNR: 23.15; MSSIM: 0.7160)

the pLATV-algorithm is solved by the semismooth Newton method described in Appendix. For the SA-TV method, we use the parameters suggested in [58] and hence  $\mathcal{I}_{i,j} = \Omega_{i,j}^\omega$ . Moreover, we set  $\lambda_0 = 0.2$  in our experiments, which seems sufficiently small. In Tables 6 and 7, 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 out-



**Table 6** PSNR- and MSSIM-values of the reconstruction of different images corrupted by salt-and-pepper noise with  $r_1 = r_2$  via pAPS- and pLATV-algorithm with  $\alpha_0 = 10^{-2}$  and SA-TV-algorithm with  $\lambda_0 = 0.2$

and window-size  $21 \times 21$ . In the pLATV-algorithm, we use  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$  with window-size  $11 \times 11$  pixels in the interior and we set  $p_0 = \frac{1}{2}$

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

A bold values indicate the best value for the corresponding experiment

**Table 7** PSNR- and MSSIM-values of the reconstruction of different images corrupted by random-valued impulse noise via pAPS- and pLATV-algorithm with  $\alpha_0 = 10^{-2}$  and SA-TV-algorithm with  $\lambda_0 = 0.2$

and window-size  $21 \times 21$ . In the pLATV-algorithm, we use  $\mathcal{I}_{i,j} = \tilde{\Omega}_{i,j}$  with window-size  $11 \times 11$  pixels in the interior and we set  $p_0 = \frac{1}{2}$  in the pLATV-algorithm

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

A bold values indicate the best value for the corresponding experiment

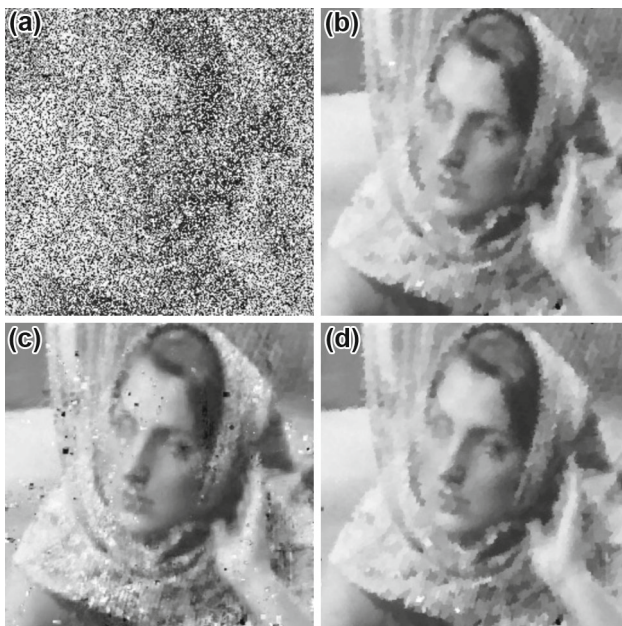
performed in most examples. For example, in Fig. 15 we observe that the pAPS- and pLATV-algorithm remove the noise considerably, while the solution of the SA-TV method still contains noise. On the contrary, for the removal of random-valued impulse noise in Fig. 16 we see that all three methods produce similar restorations.

### 7 Conclusion and Extensions

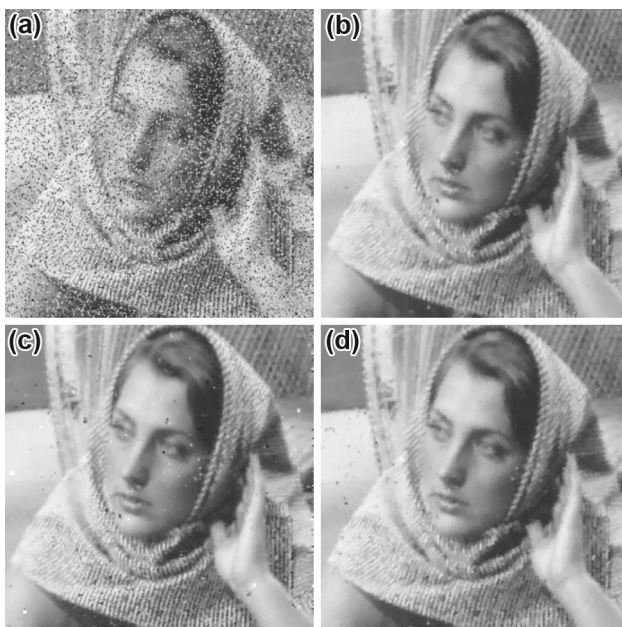
For  $L^1$ -TV and  $L^2$ -TV minimization including convolution type of problems, automatic parameter selection algorithms

for scalar and locally dependent weights  $\alpha$  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.

In order to treat homogeneous regions differently than fine features in images, which promises a better reconstruction, cf. Proposition 4.2 and Remark 4.1, algorithms for automatically computing locally adapted weights  $\alpha$  are proposed.



**Fig. 15** Reconstruction of the barbara-image corrupted by salt-and-pepper noise with  $r_1 = r_2 = 0.3$ . **a** Noisy image, **b** pAPS (PSNR: 21.56; MSSIM: 0.6242), **c** SA-TV (PSNR: 20.54; MSSIM: 0.5889), and **d** pLATV (PSNR: 21.51; MSSIM: 0.6253)



**Fig. 16** Reconstruction of the barbara-image corrupted by random-valued impulse noise with  $r = 0.3$ . **a** Noisy image, **b** pAPS (PSNR: 24.24; MSSIM: 0.8040), **c** SA-TV (PSNR: 23.96; MSSIM: 0.7977), and **d** pLATV (PSNR: 24.24; MSSIM: 0.7992)

These methods are much more stable with respect to the initial  $\alpha_0$  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 the results with scalar parameters. On the contrary, for removing impulse noise a spatially varying  $\alpha$  or  $\lambda$  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.

Although the proposed parameter selection algorithms are constructed to estimate the parameter  $\alpha$  in (1.6) and (1.9), they can be easily adjusted to find a good candidate  $\lambda$  in (1.7) and (1.10), respectively, as well.

Note that the proposed parameter selection algorithms are not restricted to total variation minimization, but may be extended to other types of regularizers as well by imposing respective assumptions that guarantee a minimizer of the considered optimization problem. In order to obtain similar (convergence) results as presented in Sects. 3 and 4, the considered regularizer should be convex, lower semicontinuous, and one-homogeneous. In particular for proving convergence results as in Sect. 3 (cf. Theorem 3.1 and Theorem 3.2), an equivalence of the penalized and corresponding constrained minimization problem, as in Theorem 2.2, is needed. An example of such a regularizer for which the presented algorithms are extendable is the total generalized variation [13].

**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 [58].

## Appendix: Semismooth Newton Method for Solving (5.9)

A semismooth Newton algorithm for solving (5.9) can be derived similar as in [58] by means of vector-valued variables. Therefore, let  $u^h \in \mathbb{R}^N$ ,  $q^h \in \mathbb{R}^{2N}$ ,  $\alpha^h \in \mathbb{R}^N$ ,  $g^h \in \mathbb{R}^N$  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  $\text{sign}(\cdot)$  are understood for vectors in a componentwise sense. Moreover, we use the function  $[\![ \cdot ]\!] : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$  with  $[\![v^h]\!]_i = [\![v^h]\!]_{i+N} = \sqrt{(v_i^h)^2 + (v_{i+N}^h)^2}$  for  $1 \leq i \leq N$ .

For solving (5.9) in every step of our Newton method we need to solve

$$\begin{pmatrix} A_k^h & -D^h(m_{\beta_k}) & 0 \\ -\frac{1}{\beta+\mu}(T^h)^t T^h + \kappa \Delta^h & -\frac{\mu}{\mu+\beta}(T^h)^t & -(\nabla^h)^t \\ B_k^h & 0 & D^h(m_{\gamma_k}) \end{pmatrix} \begin{pmatrix} \delta_u \\ \delta_v \\ \delta_q \end{pmatrix} = \begin{pmatrix} -\mathfrak{F}_1^k \\ -\mathfrak{F}_2^k \\ -\mathfrak{F}_3^k \end{pmatrix} \tag{7.1}$$

where

$$\begin{aligned} A_k^h &= \left[ D^h(e_N) - D^h(v_k^h) \chi_{\mathcal{A}_{\beta_k}} D^h(\text{sign}(T^h u_k^h - g^h)) \right] T^h, \\ B_k^h &= \left[ D^h(q_k^h) \chi_{\mathcal{A}_{\gamma_k}} D^h(m_{\gamma_k})^{-1} M^h(\nabla^h u_k^h) \right. \\ &\quad \left. - D^h((\alpha^h, \alpha^h)^t) \right] \nabla^h, \\ \mathfrak{F}_1^k &= T u_k^h - g^h - D^h(m_{\beta_k}) v_k^h, \\ \mathfrak{F}_2^k &= -(\nabla^h)^t q_k^h + \kappa \Delta^h u_k^h - \frac{1}{\beta + \mu} (T^h)^t (T^h u_k^h - g^h) \\ &\quad - \frac{\mu}{\mu + \beta} (T^h)^t v_k^h, \\ \mathfrak{F}_3^k &= -D^h \left( (\alpha^h, \alpha^h)^t \right) \nabla^h u_k^h + D^h(m_{\gamma_k}) q_k^h, \end{aligned}$$

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

$$\begin{aligned} \chi_{\mathcal{A}_{\beta_k}} &= D^h(t_{\beta_k}) \quad \text{with } (t_{\beta_k})_i = \begin{cases} 0 & \text{if } (m_{\beta_k})_i = \beta, \\ 1 & \text{else;} \end{cases} \\ \chi_{\mathcal{A}_{\gamma_k}} &= D^h(t_{\gamma_k}) \quad \text{with } (t_{\gamma_k})_i = \begin{cases} 0 & \text{if } (m_{\gamma_k})_i = \gamma(\alpha^h)_i, \\ 1 & \text{else;} \end{cases} \\ M^h(v) &= \begin{pmatrix} D^h(v_x) & D^h(v_y) \\ D^h(v_x) & D^h(v_y) \end{pmatrix} \quad \text{with } v = (v_x, v_y)^t \in \mathbb{R}^{2N}. \end{aligned}$$

Since the diagonal matrices  $D^h(m_{\beta_k})$  and  $D^h(m_{\gamma_k})$  are invertible, we eliminate  $\delta_v$  and  $\delta_q$  from (7.1), which leads to the following resulting system:

$$H_k \delta_u = f_k$$

where

$$\begin{aligned} H_k &:= \frac{1}{\beta + \mu} (T^h)^t T^h - \kappa \Delta^h + \frac{\mu}{\mu + \beta} (T^h)^t D^h(m_{\beta_k})^{-1} A_k^h \\ &\quad + (\nabla^h)^t D^h(m_{\gamma_k})^{-1} (-B_k^h), \\ f_k &:= \mathfrak{F}_2^k - \frac{\mu}{\mu + \beta} (T^h)^t D^h(m_{\beta_k})^{-1} \mathfrak{F}_1^k \\ &\quad + (\nabla^h)^* D^h(m_{\gamma_k})^{-1} \mathfrak{F}_3^k. \end{aligned}$$

In general,  $B_k^h$  and hence  $H_k$  is not symmetric. In [59], it is shown that the matrix  $H_k^h$  at the solution  $(u_k^h, v_k^h, q_k^h) = (\bar{u}, \bar{v}, \bar{q})$  is positive definite whenever

$$|[q_k^h]_i| \leq (\alpha^h)_i \quad \text{and} \quad |(v_k^h)_i| \leq 1 \tag{7.2}$$

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 [36]. As pointed out in [58], we may use  $H_k^+ + \varepsilon_k D^h(e_N)$  with  $\kappa = 0$ ,  $\varepsilon_k > 0$  and  $\varepsilon_k \downarrow 0$  as  $k \rightarrow \infty$  instead of  $\kappa > 0$  to obtain a positive definite matrix. Then our semismooth Newton solver may be written as in [58]:

**Semismooth 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 (7.2) is not satisfied, then compute  $H_k^+$ ; otherwise set  $H_k^+ := H_k$ .
3. Solve  $H_k^+ \delta_u = f_k$  for  $\delta_u$ .
4. Compute  $\delta_q$  by using  $\delta_u$ .
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 [52,59].

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

### References

1. 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)
2. Almansa, A., Ballester, C., Caselles, V., Haro, G.: A TV based restoration model with local constraints. *J. Sci. Comput.* **34**(3), 209–236 (2008)
3. 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)
4. Aubert, G., Aujol, J.F.: A variational approach to removing multiplicative noise. *SIAM Journal on Applied Mathematics* **68**(4), 925–946 (2008)
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. Bertalmio, 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.:  $\Gamma$ -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. Cai, J.F., Chan, R.H., Nikolova, M.: Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise. *Inverse Problems and Imaging* **2**(2), 187–204 (2008)
  16. Calatroni, L., Chung, C., De Los Reyes, J.C., Schönlieb, C.B., Valkonen, T.: Bilevel approaches for learning of variational imaging models. *arXiv preprint arXiv:1505.02120* (2015)
  17. 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)
  18. 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
  19. 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)
  20. Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. *Numer. Math.* **76**(2), 167–188 (1997)
  21. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming* pp. 1–35
  22. 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)
  23. Chan, T.F., Esedoğlu, S.: Aspects of total variation regularized  $L^1$  function approximation. *SIAM J. Appl. Math.* **65**(5), 1817–1837 (2005)
  24. 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)
  25. Chan, T.F., Shen, J., Zhou, H.M.: Total variation wavelet inpainting. *Journal of Mathematical imaging and Vision* **25**(1), 107–125 (2006)
  26. Chung, V.C., De Los Reyes, J.C., Schönlieb, C.B.: Learning optimal spatially-dependent regularization parameters in total variation image restoration. *arXiv preprint arXiv:1603.09155* (2016)
  27. 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
  28. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.* **4**(4), 1168–1200 (2005). (electronic)
  29. Darbon, J., Sigelle, M.: A fast and exact algorithm for total variation minimization. In: *Pattern recognition and image analysis*, pp. 351–359. Springer (2005)
  30. 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)
  31. 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)
  32. Daubechies, I., Teschke, G., Vese, L.: Iteratively solving linear inverse problems under general convex constraints. *Inverse Probl. Imaging* **1**(1), 29–46 (2007)
  33. De Los Reyes, J.C., Schönlieb, C.B.: Image denoising: Learning the noise model via nonsmooth PDE-constrained optimization. *Inverse Probl. Imaging* **7**(4) (2013)
  34. Deledalle, C.A., Vaiter, S., Fadili, J., Peyré, G.: Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. *SIAM Journal on Imaging Sciences* **7**(4), 2448–2487 (2014)
  35. Dobson, D.C., Vogel, C.R.: Convergence of an iterative method for total variation denoising. *SIAM J. Numer. Anal.* **34**(5), 1779–1791 (1997)
  36. 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)
  37. 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)
  38. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association* **90**(432), 1200–1224 (1995)
  39. 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
  40. Eldar, Y.C.: Generalized sure for exponential families: Applications to regularization. *Signal Processing, IEEE Transactions on* **57**(2), 471–481 (2009)
  41. Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*, *Mathematics and its Applications*, vol. 375. Kluwer Academic Publishers Group, Dordrecht (1996)
  42. Fornasier, M., Naumova, V., Pereverzyev, S.V.: Parameter choice strategies for multipenalty regularization. *SIAM Journal on Numerical Analysis* **52**(4), 1770–1794 (2014)
  43. Getreuer, P., Tong, M., Vese, L.A.: A variational model for the restoration of MR images corrupted by blur and Rician noise. In: *Advances in Visual Computing*, pp. 686–698. Springer (2011)
  44. Gilboa, G., Sochen, N., Zeevi, Y.Y.: Texture preserving variational denoising using an adaptive fidelity term. In: *Proc. VLsM*, vol. 3 (2003)
  45. 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)
  46. Giusti, E.: *Minimal Surfaces and Functions of Bounded Variation*, *Monographs in Mathematics*, vol. 80. Birkhäuser Verlag, Basel (1984)
  47. Goldstein, T., Osher, S.: The split Bregman method for  $L_1$ -regularized problems. *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009)
  48. 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)
  49. Hansen, P.C.: Analysis of discrete ill-posed problems by means of the  $L$ -curve. *SIAM Rev.* **34**(4), 561–580 (1992)

50. 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)
51. 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)
52. Hintermüller, M., Kunisch, K.: Total bounded variation regularization as a bilaterally constrained optimization problem. *SIAM Journal on Applied Mathematics* **64**(4), 1311–1333 (2004)
53. 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)
54. Hintermüller, M., Langer, A.: Adaptive regularization for Parseval frames in image processing. SFB-Report No. **2014-014**, 12 (2014)
55. Hintermüller, M., Langer, A.: Non-overlapping domain decomposition methods for dual total variation based image denoising. *Journal of Scientific Computing* **62**(2), 456–481 (2015)
56. Hintermüller, M., Rautenberg, C.: Optimal selection of the regularization function in a generalized total variation model. Part I: Modelling and theory. WIAS preprint 2235 (2016)
57. Hintermüller, M., Rautenberg, C., Wu, T., Langer, A.: Optimal selection of the regularization function in a generalized total variation model. Part II: Algorithm, its analysis and numerical tests. WIAS preprint 2236 (2016)
58. 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)
59. 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)
60. Kindermann, S., Osher, S., Jones, P.W.: Deblurring and denoising of images by nonlocal functionals. *Multiscale Model. Simul.* **4**(4), 1091–1115 (2005). (electronic)
61. Kunisch, K., Pock, T.: A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences* **6**(2), 938–983 (2013)
62. Langer, A.: Subspace correction and domain decomposition methods for total variation minimization. Ph.D. thesis, Johannes Kepler Universität Linz (2011)
63. Langer, A., Osher, S., Schönlieb, C.B.: Bregmanized domain decomposition for image restoration. *Journal of Scientific Computing* **54**(2–3), 549–576 (2013)
64. Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. *Journal of mathematical imaging and vision* **27**(3), 257–263 (2007)
65. Li, F., Ng, M.K., Shen, C.: Multiplicative noise removal with spatially varying regularization parameters. *SIAM J. Imaging Sci.* **3**(1), 1–20 (2010)
66. 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)
67. Lin, Y., Wohlberg, B., Guo, H.: UPRE method for total variation parameter selection. *Signal Processing* **90**(8), 2546–2551 (2010)
68. Lorenz, D.A., Pock, T.: An inertial forward-backward algorithm for monotone inclusions. *Journal of Mathematical Imaging and Vision* **51**(2), 311–325 (2015)
69. Mallows, C.L.: Some comments on  $C_p$ . *Technometrics* **15**(4), 661–675 (1973)
70. 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
71. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* **42**(5), 577–685 (1989)
72. Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Program.* **103**((1, Ser. A)), 127–152 (2005)
73. 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)
74. Nikolova, M.: Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM J. Numer. Anal.* **40**(3), 965–994 (2002). (electronic)
75. Nikolova, M.: A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision* **20**(1–2), 99–120 (2004)
76. 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 (2005). (electronic)
77. 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)
78. Ramani, S., Liu, Z., Rosen, J., Nielsen, J., Fessler, J.A.: Regularization parameter selection for nonlinear iterative image restoration and MRI reconstruction using GCV and SURE-based methods. *IEEE Transactions on Image Processing* **21**(8), 3659–3672 (2012)
79. Rockafellar, R.T.: *Convex Analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J (1970)
80. 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)
81. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* **60**(1), 259–268 (1992)
82. Stein, C.M.: Estimation of the mean of a multivariate normal distribution. *The annals of Statistics* pp. 1135–1151 (1981)
83. 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)
84. Strong, D.M., Chan, T.F.: Spatially and scale adaptive total variation based regularization and anisotropic diffusion in image processing. In: *Division in Image Processing, UCLA Math Department CAM Report. Citeseer* (1996)
85. 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)
86. Tadmor, E., Nezzar, S., Vese, L.: A multiscale image representation using hierarchical  $(BV, L^2)$  decompositions. *Multiscale Model. Simul.* **2**(4), 554–579 (2004). (electronic)
87. 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)
88. Tikhonov, A.N., Arsenin, V.Y.: *Solutions of Ill-Posed Problems*. Vh Winston, (1977)
89. Vogel, C.R.: Non-convergence of the  $L$ -curve regularization parameter selection method. *Inverse Problems* **12**(4), 535–547 (1996)
90. 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)
91. 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)
92. 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)
93. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report* pp. 08–34 (2008)



**Andreas Langer** studied Mathematics at the Johannes Kepler University Linz (Austria) and received his Ph.D. in Linz in 2011. From 2011 to 2014, he worked as a PostDoc at the Karl-Franzens University Graz (Austria). Currently, he is a PostDoc at the Chair of Numerical Mathematics for High Performance Computing at the University of Stuttgart (Germany). His research interests include approximation techniques for partial differential

equations, variational methods, and subspace correction methods, with applications in image processing.