



# ADMM-TGV image restoration for scientific applications with unbiased parameter choice

Christian Zietlow<sup>1</sup> · Jörg K. N. Lindner<sup>1</sup>

Received: 1 August 2023 / Accepted: 10 January 2024  
© The Author(s) 2024

## Abstract

Image restoration via alternating direction method of multipliers (ADMM) has gained large interest within the last decade. Solving standard problems of Gaussian and Poisson noise, the set of “Total Variation” (TV)-based regularizers proved to be efficient and versatile. In the last few years, the “Total Generalized Variation” (TGV) approach combined TV regularizers of different orders adaptively to better suit local regions in the image. This improved the technique significantly. The approach solved the staircase problem inherent of the first-order TV while keeping the beneficial edge preservation. The iterative minimization for the augmented Lagrangian of TGV problems requires four important parameters: two penalty parameters  $\rho$  and  $\eta$  and two regularization parameters  $\lambda_0$  and  $\lambda_1$ . The choice of penalty parameters decides on the convergence speed, and the regularization parameters decide on the impact of the respective regularizer and are determined by the noise level in the image. For scientific applications of such algorithms, an automated and thus objective method to determine these parameters is essential to receive unbiased results independent of the user. Obviously, both sets of parameters are to be well chosen to achieve optimal results, too. In this paper, a method is proposed to adaptively choose optimal  $\rho$  and  $\eta$  values for the iteration to converge faster, based on the primal and dual residuals arising from the optimality conditions of the augmented Lagrangian. Further, we show how to choose  $\lambda_0$  and  $\lambda_1$  based on the inherent noise in the image.

**Keywords** ADMM · Residual balancing · Total generalized variation · TGV · Gaussian noise · Poisson noise

**Mathematics Subject Classification (2010)** 65B99

---

✉ Christian Zietlow  
christian.zietlow@upb.de

Jörg K. N. Lindner  
lindner@physik.upb.de

<sup>1</sup> Department of Physics, Paderborn University, Warburger Str. 100, Paderborn 33098, Germany

## 1 Introduction

Every measurement contains noise, which most of the time is undesirable, as it disguises the signal and thus the information to be extracted from it [1]. Often, iterative denoising [2–6] is needed to recover part of the information hidden by the noise. For recovering as much of information as possible, the algorithm has to address the right noise statistics, obtain a suitable prior, and, of course, converge to a certain solution. One category of algorithms for denoising relies on the principles of solving the “augmented Lagrangian,” which is adaptable to a large variety of problems concerning image optimization. One of these algorithms is the “alternating direction method of multipliers” (ADMM) algorithm, which was first described by Glowinski and Marroco [7] and Gabay and Mercier [8]. It has proven to be a good tool to solve these kinds of ill-posed inverse problems successfully and thus has gained huge interest within the last decade, during which huge efforts were made to optimize specialized priors or regularizers for all types of structure occurring in images [9–15]. Many of these algorithms interpret the principle of “total variation” (TV) by Rudin et al. [16] in slightly different ways. The newest approach called “total generalized variation” (TGV) from Bredis et al. [17] utilizes an adaptive mixture of different higher orders of TV and applies them on suiting local regions on the image, solving the staircase problem of first order TV, with the benefit of preserving sharp edges, that would otherwise be lost in higher-order TV approaches [18].

As will be shown in this work, successful denoising and deconvolution in the context of total variation break down to the choice of two sets of parameters: the penalty parameters  $\rho$  and  $\eta$ , which have a great influence on the convergence speed [19–22], and the regularization parameters  $\lambda_0$  and  $\lambda_1$ , that depend on the amount of noise and determine the quality of the outcome [23, 24]. To use these ADMM-based algorithms to evaluate experimental data in a scientific context, the choice of these parameters must necessarily be objective to avoid altering the results by the user. Today, this hurdle restricts natural scientist to switch to better-suited algorithms. Instead, it pushes them to use algorithms, such as the “Richardson-Lucy-Algorithm” (RLA), Fourier-ratio deconvolution, or Wiener filtering [25–31], that cannot compete with modern algorithms, such as the ADMM, in terms of adaptability and versatility.

By balancing primal and dual residuals, arising from optimality conditions for the ADMM, the penalty parameters can be found adaptively in a stable scheme for image denoising or deconvolution, speeding up convergence and making the algorithm independent on the initial choice of these parameters. Further, we will show how regularization parameters  $\lambda_0$  and  $\lambda_1$  need to be chosen in order to improve the image quality significantly, based on the standard deviation of the noise. With our work, the ADMM-based TGV algorithm can be designed such that it operates purely on a mathematical foundation and on noise parameters that can be determined experimentally by analyzing the used detector itself. Thus, our algorithm enables experimental scientist to use ADMM-based denoising and deconvolution techniques, adapt them to their respective experiments, and improve their data evaluation significantly.

By testing sets of these four parameters  $\rho$  and  $\eta$  and  $\lambda_0$  and  $\lambda_1$  for noisy images, we can further show that regions of optimal results can be found for different noise types and different priors, allowing to compare results of the standard ADMM-TGV

algorithm with the ones of the residual balanced. In this way, we can show that the automated choice does not only objectify the parameters, but also increases the convergence speed for most initial values. In the following subsections, the noise and the fundamentals of the respective algorithms are explained in detail.

### 1.1 Gaussian noise fundamentals

Whenever images are captured, the acquisition process adds noise. Gaussian noise is inflicted in terms of, e.g., camera read-out noise and can be described as an additive process. The image formation process equates the following [16]:

$$\xi = \Omega \mathbf{x} + \mathcal{N}(\mu, \sigma^2), \tag{1}$$

with the measured image  $\xi$ , a point-spread-function (PSF)  $\Omega$  convolved with the noise-free image  $\mathbf{x}$  and Gaussian noise  $\mathcal{N}$  with the mean value  $\mu$  and the variance  $\sigma^2$ . To clarify notations, in the following, we will use bold letters to indicate vectors or matrices. Scalars and indices remain in normal font.

The mean value can be leveled by subtracting a background image. For removing noise with the variance  $\sigma^2$  from the data, denoising is required. In terms of Bayes' theorem [32], the likelihood  $L_{\mathcal{N}}$  of having a good estimate of the “true” convolved image  $\Omega \mathbf{x}$  can be expressed as the conditional probability  $P$  knowing  $\xi$  [33]:

$$\begin{aligned} L_{\mathcal{N}}(\Omega \mathbf{x}) &= P(\xi | \Omega \mathbf{x}) \\ &= \prod_{m,n}^{M,N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{([\Omega \mathbf{x}]_{m,n} - \xi_{m,n})^2}{2\sigma^2} \right\}, \end{aligned} \tag{2}$$

with the individual pixels  $m, n$  of an  $M \times N$  image. This equation can be reformulated into the negative log-likelihood functional [33]:

$$\begin{aligned} F_{\mathcal{N}}(\Omega \mathbf{x}) &\propto -\ln \{ L_{\mathcal{N}}(\Omega \mathbf{x}) \} \propto \sum_{m,n}^{M,N} \frac{([\Omega \mathbf{x}]_{m,n} - \xi_{m,n})^2}{2\sigma^2} \\ &= \frac{1}{2\sigma^2} \| \Omega \mathbf{x} - \xi \|_F^2, \end{aligned} \tag{3}$$

with the Frobenius norm  $\| \cdot \|_F$  as an extension of the Euclidean norm for matrices.

### 1.2 Poisson noise fundamentals

Due to the quantized nature of photons or electrons used for imaging, the Poisson noise occurs as count fluctuations. It thus differs from the Gaussian noise in the overall image statistics, as it is deeply embedded in the signal [34]:

$$\xi = \mathcal{P}(\Omega \mathbf{x}), \tag{4}$$

with the Poisson noise  $\mathcal{P}$ . Similar to flipping a coin, a probability of counting a certain number of incidents can be assigned to every pixel. Utilizing Bayes' theorem again to have the true convolved image  $\Omega\mathbf{x}$ , the likelihood can be expressed as follows [34]:

$$L_{\mathcal{P}}(\Omega\mathbf{x}) = P(\xi | \Omega\mathbf{x}) = \prod_{n,m} \frac{[\Omega\mathbf{x}]_{n,m}^{\xi_{n,m}} \exp\{-[\Omega\mathbf{x}]_{n,m}\}}{\xi_{n,m}!}, \tag{5}$$

with the negative log-likelihood functional [34]:

$$G_{\mathcal{P}}(\Omega\mathbf{x}) \propto -\ln\{L_{\mathcal{P}}(\Omega\mathbf{x})\} \tag{6}$$

$$= \sum_{n,m}^{N,M} -\ln\{[\Omega\mathbf{x}]_{n,m}\} (\xi_{n,m}) + [\Omega\mathbf{x}]_{n,m}. \tag{7}$$

In the formulation of  $G_{\mathcal{P}}$ , the factorial in the denominator can be left out, as it vanishes in the derivation and thus plays no further role in the minimization.

### 1.3 General scaled ADMM

The general scaled ADMM algorithm solves problems in the form of [22]:

$$\begin{aligned} &\arg \min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ &\text{subject to } \mathbf{Ax} + \mathbf{Bz} = \mathbf{c}, \end{aligned} \tag{8}$$

with two convex functions  $f$  and  $g$  applied to the variables  $\mathbf{x}$  and  $\mathbf{z}$ .  $\mathbf{A}$  and  $\mathbf{B}$  then are operators applied to the variables, and  $\mathbf{c}$  is a constant.

As an example in imaging,  $f$  and  $g$  can be functions that correspond to a given noise statistic or allow to distinguish between a “true” image without blur and noise and the noise itself. The variable  $\mathbf{x}$  in this case corresponds to the “true” image and  $\mathbf{z}$  to the noise to be separated. Both variables are connected via one or several conditions, including the operators  $\mathbf{A}$  and  $\mathbf{B}$  that for example could act as a convolution with a blur kernel or difference operators. The application of ADMM for image treatments is shown in Sect. 2.

Staying with more general expressions, the overall problem can be formulated into the scaled augmented Lagrangian to be minimized, which is given by the following [22]:

$$\mathcal{L}_{\rho}(\mathbf{x}, \mathbf{z}, \mathbf{u}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Ax} + \mathbf{Bz} - \mathbf{c} + \mathbf{u}\|_2^2 - \frac{\rho}{2} \|\mathbf{u}\|_2^2, \tag{9}$$

with the Euclidean norm  $\|\cdot\|_2$ , the scaled dual variable (or Lagrange multiplier)  $\mathbf{u} = \frac{1}{\rho}\mathbf{y}$  associated with the constraint  $\mathbf{Ax} + \mathbf{Bz} = \mathbf{c}$ , and  $\mathbf{y}$  being the unscaled dual

variable. The iterative minimization process then can be decomposed and carried out in a three-step pattern [22]:

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{Ax} + \mathbf{Bz}^k - \mathbf{c} + \mathbf{u}^k\|_2^2 \right\}, \tag{10}$$

$$\mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} \left\{ g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Ax}^{k+1} + \mathbf{Bz} - \mathbf{c} + \mathbf{u}^k\|_2^2 \right\}, \tag{11}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{Ax}^{k+1} + \mathbf{Bz}^{k+1} - \mathbf{c}, \tag{12}$$

minimizing with respect to  $\mathbf{x}$  first,  $\mathbf{z}$  second and then updating the dual variable  $\mathbf{u}$ . In this context,  $\rho > 0$  appears as a penalty parameter. Its careful choice is crucial for the convergence speed, as a good choice might result in an accurate solution within a few tens of iterations, while a bad choice might delay convergence to a couple of thousand iterations or even lead to divergence. Obviously, it is desirable to minimize time cost, so somehow, an optimal  $\rho$  choice has to be found.

### 1.4 Optimality conditions and residual balancing

Three feasibility conditions have to be fulfilled for the ADMM to optimally converge. These arise from the Karush-Kuhn-Tucker conditions [35], which utilize first-order derivatives to find a saddle point in the Lagrangian. Deriving  $\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u})$  with respect to the unscaled Lagrange parameter  $\mathbf{y}$  yields the primal feasibility condition [9, 22]:

$$0 = \mathbf{Ax}^* + \mathbf{Bz}^* - \mathbf{c}, \tag{13}$$

with the star marking the optimum choice of the respective variable. As the Lagrangian is minimized by the updates  $\mathbf{x}^{k+1}$  and  $\mathbf{z}^{k+1}$ , the difference from zero of the above equation gives the primal residual [9, 22]:

$$\mathbf{R}^{k+1} = \mathbf{Ax}^{k+1} + \mathbf{Bz}^{k+1} - \mathbf{c}. \tag{14}$$

Second, both the  $\mathbf{x}$ - and  $\mathbf{z}$ -updates must satisfy the dual feasibility conditions with respect to the unscaled dual variable  $\mathbf{y} = \rho\mathbf{u}$ , yielding the following [9, 22]:

$$0 \in \partial_{\mathbf{x}} f(\mathbf{x}^*) + \rho\mathbf{A}^T \mathbf{u}^*, \tag{15}$$

$$0 \in \partial_{\mathbf{z}} g(\mathbf{z}^*) + \rho\mathbf{B}^T \mathbf{u}^*, \tag{16}$$

with  $\mathbf{A}^T, \mathbf{B}^T$  denoting the adjoint operator of  $\mathbf{A}$  and  $\mathbf{B}$ . The subdifferential operator  $\partial$  can be replaced by a gradient and the  $\in$  by an equal sign, in the case of  $g$  and  $f$  being differentiable [22]. Further, assuming  $\rho^k$  to be variable between iterations, where  $\tau^k$  optimizes  $\rho^{k+1}$ :

$$\rho^{k+1} = \tau^k \cdot \rho^k, \tag{17}$$

$$\mathbf{u}^{k+1} = \frac{\mathbf{y}^k}{\rho^{k+1}} = \frac{\mathbf{u}^k + \mathbf{R}^{k+1}}{\tau^k}. \tag{18}$$

Then, the scaled dual variable has to be rescaled inversely [22], to adjust for the right  $\mathbf{y}^{k+1}$ . To make it easier to follow the general idea, we switch the order of both dual feasibility conditions for a moment. The approach for the residual balancing is hidden in (15), eventually. However, (16) is needed for completeness:

As the  $\mathbf{z}$ -update happens after the  $\mathbf{x}$ -update,  $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k$  is already updated. Per definition, the  $\mathbf{z}$ -update  $\mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{u}^k)$  is minimized by  $\mathbf{z}^{k+1}$ , and deriving with respect to  $\mathbf{z}$  and substituting it into the equation (16) yield the following [22]:

$$\begin{aligned} 0 &\in \partial_{\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}^{k+1}, \mathbf{u}^k) \\ &= \partial_{\mathbf{z}} g(\mathbf{z}^{k+1}) + \rho^k \mathbf{B}^T \mathbf{u}^k + \rho^k \mathbf{B}^T (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}) \\ &= \partial_{\mathbf{z}} g(\mathbf{z}^{k+1}) + \rho^{k+1} \mathbf{B}^T \mathbf{u}^{k+1}, \end{aligned} \tag{19}$$

showing that optimality is always assured by  $\mathbf{z}^{k+1}$  and  $\mathbf{u}^{k+1}$ , even when  $\rho$  is changing. Thus, it is left to show how the  $\mathbf{x}$ -update satisfies (15). Since the  $\mathbf{x}$ -update comes before the  $\mathbf{z}$ -update,  $\mathbf{z}^k$  has not been updated yet. As per definition, the  $\mathbf{x}$ -update  $\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}^k, \mathbf{u}^k)$  is minimized by  $\mathbf{x}^{k+1}$ , and deriving with respect to  $\mathbf{x}$  and substituting it into the equation (15) yield the following [22]:

$$\begin{aligned} 0 &\in \partial_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{u}^k) \\ &= \partial_{\mathbf{x}} f(\mathbf{x}^{k+1}) + \rho^k \mathbf{A}^T \mathbf{u}^k + \rho^k \mathbf{A}^T (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^k - \mathbf{c}) \\ &= \partial_{\mathbf{x}} f(\mathbf{x}^{k+1}) + \rho^{k+1} \mathbf{A}^T \mathbf{u}^{k+1} + \rho^k \mathbf{A}^T \mathbf{B} (\mathbf{z}^k - \mathbf{z}^{k+1}) \\ &\Leftrightarrow \rho^k \mathbf{A}^T \mathbf{B} (\mathbf{z}^{k+1} - \mathbf{z}^k) \in \partial_{\mathbf{x}} f(\mathbf{x}^{k+1}) + \rho^{k+1} \mathbf{A}^T \mathbf{u}^{k+1}. \end{aligned} \tag{20}$$

In order to assure optimality, the first term, which can be seen as the dual residual [9, 22]

$$\mathbf{S}^{k+1} = \rho^k \mathbf{A}^T \mathbf{B} (\mathbf{z}^{k+1} - \mathbf{z}^k), \tag{21}$$

has to vanish. Further, Wohlberg suggested the usage of relative residuals [36] making them invariant to problem scaling. In contrast to his approach, we propose to alter the normalization of the relative primal residual  $\mathbf{R}_{rel}^{k+1}$  slightly, such that  $\mathbf{c}$  is grouped with the  $\mathbf{B}\mathbf{z}^{k+1}$  norm, making the norm more resilient against high constants  $\mathbf{c}$ :

$$\mathbf{S}_{rel}^{k+1} = \frac{\rho^k \mathbf{A}^T \mathbf{B} (\mathbf{z}^{k+1} - \mathbf{z}^k)}{\max \{ \|\partial_{\mathbf{x}} f(\mathbf{x}^{k+1})\|_2, \|\rho^k \mathbf{A}^T (\mathbf{u}^k + \mathbf{R}^{k+1})\|_2 \}}, \tag{22}$$

$$\mathbf{R}_{rel}^{k+1} = \frac{\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}}{\max \{ \|\mathbf{A}\mathbf{x}^{k+1}\|_2, \|\mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}\|_2 \}}. \tag{23}$$

He et al. [37] define the distance from convergence for ADMM based on these residuals as  $D^{k+1} = \|\mathbf{R}^{k+1}\|_2^2 + \|\mathbf{S}^{k+1}\|_2^2$  allowing to increase the performance of the algorithm. Equations (10) and (11) suggest a decreasing norm of the primal residual for an increasing  $\rho$ , but (21) suggests an increase of dual residual norm simultaneously—for decreasing  $\rho$  vice versa. So, somehow, balancing  $\rho$ , such that

both residuals approach zero, seems to be a reasonable heuristic for minimizing the overall distance [36]. As both norms for the relative residuals converge, a similar relative distance measure  $D_{rel}^{k+1} = \|\mathbf{R}_{rel}^{k+1}\|_2^2 + \|\mathbf{S}_{rel}^{k+1}\|_2^2$  must also vanish, showing that the normalization of the residuals is also a valid strategy.

For the purpose of utilizing both residuals, Boyd proposes an update scheme for  $\rho$ , which keeps both residuals in the same magnitude, called “residual balancing” [22]:

$$\rho^{k+1} = \begin{cases} \rho^k \cdot \tau & \text{if } \|\mathbf{R}^{k+1}\|_2 > \mu \|\mathbf{S}^{k+1}\|_2 \\ \rho^k / \tau & \text{if } \|\mathbf{S}^{k+1}\|_2 > \mu \|\mathbf{R}^{k+1}\|_2 \\ \rho^k & \text{otherwise} \end{cases} \quad (24)$$

with a balancing parameter  $\mu = 10$  and  $\tau = 2$ . This approach was found successful by various authors [38–41]. In the proof by He et al. [37], convergence for this adaptive scheme comes down to bounds on the sequence  $\rho^k$  [37]:

$$\begin{aligned} \inf_k \{\rho^k\} > 0 \quad \text{and} \quad \sup_k \{\rho^k\} < +\infty, & (25) \\ \text{with } \sum_{k=0}^{\infty} \tau_k - 1 < +\infty & . \end{aligned} \quad (26)$$

The latter is a slight variation of the original proof, as He et al. define their factor (here  $\hat{\tau}$ ) slightly different than the authors do  $\tau = (\hat{\tau} + 1)$ . The main consequence of this proof is that  $\tau \rightarrow 1$  must converge eventually. Since the definition  $\tau > 1$  is necessary for  $\tau$  to be useful, it increases the sum (26) every iteration indefinitely until it converges to 1. In Boyd’s approach, the infinite sum is mainly prevented by a high value of  $\mu$ , hoping that at some point,  $\mathbf{R}$  and  $\mathbf{S}$  diverge so slowly from one another, that this threshold basically stops any further updates from happening. A low value of  $\mu$  in contrast would be desirable, as it would adjust  $\rho$  much faster. However, a low  $\mu$  leads to balancing  $\rho^k$  regularly, making it oscillate around an optimal  $\rho$  and never attaining it. This necessarily violates (26) for any value  $\tau \neq 1$ , which, despite converging, would be pointless. This scheme is thus compelled to updating  $\rho$  quite slowly, as it takes time for  $\mathbf{R}$  and  $\mathbf{S}$  to diverge from one another.

Both residuals can be utilized in a more direct way, as Wohlberg showed by using relative residuals, extending the scheme to choosing  $\tau = \tau^k$  at each iteration and utilizing a ratio of the relative residuals  $v^{k+1}$  to make the  $\rho$  update much faster [36]:

$$\tau^{k+1} = \begin{cases} v^{k+1} & \text{if } \tau_{max} \geq v^{k+1} \geq 1 \\ (v^{k+1})^{-1} & \text{if } \tau_{max}^{-1} \leq v^{k+1} < 1 \\ \tau_{max} & \text{otherwise} \end{cases} \quad (27)$$

$$v^{k+1} = \sqrt{\|\mathbf{R}_{rel}^{k+1}\|_2 / \|\mathbf{S}_{rel}^{k+1}\|_2}.$$

Since  $\tau_{max}$  provides a bound on  $\tau$ , convergence still holds for the extension, as the sequence  $\rho^k$  would be bounded away from zero and infinity [36]. But again,  $\tau$  must eventually settle to 1 or updates must unnecessarily be delayed by a high value of  $\mu$ .

However, the general idea of delaying the updates is not wrong, but the method of thresholding the residuals by a factor  $\mu$  proves to be unpredictable. Since the initial choice of  $\rho$  determines the development of both primal and dual residual, respectively, it depends on that very choice whether the first balancing of  $\rho$  appears sooner or later during the iterations. But since the progress in the early iterations is much larger than in the later ones, it is important to place  $\rho$  on the right track, as early as possible.

Here, we propose a similar method with neither the strict requirement  $\tau \rightarrow 1$  nor the unpredictability. By choosing  $\mu = 1$ , the above extended scheme (24) with (27) can be simplified to the following:

$$\tau^{k+1} = \begin{cases} \nu^{k+1} & \text{if } 0 < \nu^{k+1} < \infty \\ \min \{ \rho^{-1}, \tau_{ext} \} & \text{elseif } \rho < 1 \\ \max \{ \rho^{-1}, \tau_{ext}^{-1} \} & \text{elseif } \rho > 1 \\ 1 & \text{otherwise} \end{cases} \quad (28)$$

$$\rho^{k+1} = \begin{cases} \rho^k \cdot \tau^{k+1} & \text{if } \rho_{max}^{-1} \leq \rho^k \cdot \tau^{k+1} \leq \rho_{max} \\ \rho_{max}^{-1} / \rho^k & \text{if } \rho_{max}^{-1} < \rho^k \cdot \tau^{k+1} \\ \rho_{max} / \rho^k & \text{if } \rho^k \cdot \tau^{k+1} > \rho_{max} \end{cases} \quad (29)$$

We propose to balance  $\rho$  by the above equations, but to increase the length of the update interval exponentially. This could be done by, e.g., updating ten times per magnitude of iterations, such that the updates are exponentially spaced:

$$\tau^{k+1} = \begin{cases} \tau^{k+1} & \text{if } k \in \text{Val} \\ 1 & \text{otherwise} \end{cases}, \quad (30)$$

where Val is a series of exponentially growing values. Thus, the update occurs regularly in the early iterations, where it is necessary to find a good  $\rho$ , and fades out at a later stage, when convergence is more important. With the exponential increase of the update interval, the interval length eventually approaches infinity, which is equal to the standard ADMM, assuring convergence. Since  $\tau^k = 1$  for every step without balancing, the mean value of all  $\tau^k$  between two balancing events approaches one, which satisfies (26).

In the above equations,  $\rho$  is bounded away from zero and infinity by forbidding a direct scaling with these and by limiting the number of update steps. However,  $\nu^{k+1}$  being either zero or infinity indicates an over-extension of the balancing parameter  $\rho$ . A useful strategy is to relieve an overextended  $\rho$  by pushing the parameter back towards 1 by introducing a factor  $\tau_{ext} > 1$ , with which  $\rho$  is either increased or decreased monotonically until  $\nu^{k+1} \in (0, \infty)$  again is within boundaries and is well defined. For both, increasing or decreasing  $\rho$  monotonically He et al. proved convergence [37]. Further, the authors suggest to set a limit  $\rho_{max}$  to assure the numerical stability of the algorithm.



## 2 ADMM with total generalized variation

In order to distinguish “true” image  $\mathbf{x}$  and noise, some additional information about the image needs to be taken into consideration. The total variation approach relies on the assumption that images are smooth to a certain degree and noise is highly volatile. Consequently, derivatives can be utilized to find the noise, as the derivative of the noise is expected a lot higher than the derivative of the smooth and slowly changing image. However, if the image has a gradient itself, this assumption is inconvenient as part of the image slope is forced to be piece-wise constant in the reconstruction, leading to staircase artifacts [18]. Here, a higher-order derivative can help to smoothen the gradient of the image, instead of the image itself, leading to a better approximation of the image, but at the cost of preserving sharp edges.

Total generalized variation of second order as a method for image smoothing was first described by Bredis et al. [17] aiming to combine both orders of derivatives in a beneficial way, such that regions of the image are denoised in the most optimal manner with the help of self-adapting filter masks.

In terms of an isotropic TGV penalizer, an ADMM algorithm can be formulated following Shirai and Okuda [42]:

$$\|\mathbf{x}\|_{TGV}^{\lambda_0, \lambda_1} := \min_{\mathbf{D}\mathbf{x}=\mathbf{s}+\mathbf{t}} (\lambda_0 \|\mathbf{s}\|_F + \lambda_1 \|\mathbf{G}\mathbf{t}\|_F) . \tag{31}$$

The auxiliary matrices  $\mathbf{s}$  and  $\mathbf{t}$  are linked to the flat and sloped region of the image, where  $\mathbf{t}$  contains the latter. The matrices  $\mathbf{D}$  and  $\mathbf{G}$  contain derivatives in all possible directions. For images, they are given as follows [42]:

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_h \\ \mathbf{D}_v \end{pmatrix} , \quad \mathbf{G}^T = \begin{pmatrix} \mathbf{D}_h & \mathbf{D}_v & \mathbb{0} \\ \mathbb{0} & \mathbf{D}_h & \mathbf{D}_v \end{pmatrix} , \tag{32}$$

where the individual entries  $\mathbf{D}_h$  and  $\mathbf{D}_v$  are forward difference operators for the horizontal and vertical direction. Both and the matrix of zeros  $\mathbb{0}$  follow a “block circulant with circulant blocks” (BCCB) structure [42]. So for an image with  $N \times M$  pixels, all entries  $\mathbf{D}_{h,v}$  and the matrix of zeros  $\mathbb{0}$  have the same size and dimension as said image  $\{\mathbf{D}_h, \mathbf{D}_v, \mathbb{0}\} \in \mathbb{R}^{N \times M}$ . According to (31), the minimization separates the image in flat and sloped regions, which later serve as filter masks for first- and second-order smoothing. Thus,  $\mathbf{t}$  contains the first-order derivatives of the sloped regions of the image, at which a further derivation  $\mathbf{G}$  is applied. These sloped regions get subtracted from  $\mathbf{D}\mathbf{x}$  leaving over the first-order filter mask  $\mathbf{s}$ , which then only contains the flat regions of the image.

### 2.1 Gaussian noise ADMM with TGV

Depending on the noise type, the structure of the ADMM must be adapted. For the Gaussian-type noise, following (3), it can be formulated as follows [42]:

$$\arg \min_{\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2} 1/(2\sigma^2) \|\mathbf{\Omega}\mathbf{x} - \boldsymbol{\xi}\|_F^2 + \lambda_0 \|\mathbf{z}_1\|_F + \lambda_1 \|\mathbf{z}_2\|_F$$

$$\text{subject to} \quad \mathbf{z}_1 = \mathbf{D}\mathbf{x} - \mathbf{t}, \mathbf{z}_2 = \mathbf{G}\mathbf{t} \quad , \quad (33)$$

with the acquired image  $\xi$ , a point-spread-function  $\Omega$ , and the regularization parameters  $\lambda_0$  and  $\lambda_1$ . The augmented Lagrangian to be solved is given as follows [42]:

$$\begin{aligned} \mathcal{L}_{\rho,\eta}(\mathbf{x}, \mathbf{t}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{u}_1, \mathbf{u}_2) = & 1/(2\sigma^2) \|\Omega\mathbf{x} - \xi\|_F^2 \\ & + \lambda_0 \|\mathbf{z}_1\|_F + \rho/2 \|\mathbf{D}\mathbf{x} - \mathbf{t} - \mathbf{z}_1 + \mathbf{u}_1\|_F^2 - \rho/2 \|\mathbf{u}_1\|_F^2 \\ & + \lambda_1 \|\mathbf{z}_2\|_F + \eta/2 \|\mathbf{G}\mathbf{t} - \mathbf{z}_2 + \mathbf{u}_2\|_F^2 - \eta/2 \|\mathbf{u}_2\|_F^2. \end{aligned} \quad (34)$$

with the penalty parameters  $\rho$  and  $\eta$ . Note that for pure denoising problems, the “deconvolution kernel”  $\Omega$  simplifies to a matrix of ones  $\Omega = \mathbb{1}$ . Further, it is important to mention that often in literature,  $\sigma^2$  is found to be absorbed into  $\hat{\rho} = \rho\sigma^2$  and  $\hat{\eta} = \eta\sigma^2$  and thus excluded from the equation. But to show the dependency of the optimal  $\lambda \propto 1/\sigma$ , it is more convenient to stay with the definition chosen here.

Starting with  $\mathbf{x}^0 = \mathbb{0}$ ,  $\mathbf{t}_i^0 = \mathbb{0}$ ,  $\mathbf{z}_{1,i}^0 = \mathbb{0}$ ,  $\mathbf{z}_{2,j}^0 = \mathbb{0}$ ,  $\mathbf{u}_{1,i}^0 = \mathbb{0}$  and  $\mathbf{u}_{2,j}^0 = \mathbb{0}$ , with  $i$  and  $j$  denoting the respective directions of the derivatives and  $\mathbb{0}$  being matrices of zeros, the iteration steps are given as follows [42]:

$$\begin{aligned} \{\mathbf{x}^{k+1}, \mathbf{t}^{k+1}\} &= \arg \min_{\mathbf{x}, \mathbf{t}} \mathcal{L}_{\rho,\eta}(\mathbf{x}, \mathbf{t}, \mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{u}_1^k, \mathbf{u}_2^k) \quad , \\ \mathbf{z}_{1,i}^{k+1} &= \arg \min_{\mathbf{z}_{1,i}} \mathcal{L}_{\rho}(\mathbf{x}^{k+1}, \mathbf{t}_i^{k+1}, \mathbf{z}_{1,i}, \mathbf{u}_{1,i}^k) \quad , \\ \mathbf{z}_{2,j}^{k+1} &= \arg \min_{\mathbf{z}_{2,j}} \mathcal{L}_{\eta}(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_{2,j}, \mathbf{u}_{2,j}^k) \quad , \\ \mathbf{u}_{1,i}^{k+1} &= \mathbf{u}_{1,i}^k + \left( [\mathbf{D}\mathbf{x}^{k+1}]_i - \mathbf{t}_i^{k+1} - \mathbf{z}_{1,i}^{k+1} \right) \quad , \\ \mathbf{u}_{2,j}^{k+1} &= \mathbf{u}_{2,j}^k + \left( [\mathbf{G}\mathbf{t}^{k+1}]_j - \mathbf{z}_{2,j}^{k+1} \right) \quad . \end{aligned} \quad (35)$$

For images  $i \in \{h, v\}$  and  $j \in \{h, d, v\}$ , with  $d$  denoting the diagonal direction connected to both horizontal and vertical derivatives. Further,  $\mathbf{t}$ ,  $\mathbf{z}_{1,2}$ , and  $\mathbf{u}_{1,2}$  are given as follows [42]:

$$\begin{aligned} \mathbf{t} &= \begin{pmatrix} \mathbf{t}_h \\ \mathbf{t}_v \end{pmatrix} \quad , \\ \mathbf{z}_1 &= \begin{pmatrix} \mathbf{z}_{1,h} \\ \mathbf{z}_{1,v} \end{pmatrix} \quad , \quad \mathbf{u}_1 = \begin{pmatrix} \mathbf{u}_{1,h} \\ \mathbf{u}_{1,v} \end{pmatrix} \quad , \\ \mathbf{z}_2 &= \begin{pmatrix} \mathbf{z}_{2,h} \\ \mathbf{z}_{2,d} \\ \mathbf{z}_{2,v} \end{pmatrix} \quad , \quad \mathbf{u}_2 = \begin{pmatrix} \mathbf{u}_{2,h} \\ \mathbf{u}_{2,d} \\ \mathbf{u}_{2,v} \end{pmatrix} \quad , \end{aligned} \quad (36)$$

with, according to the BCCB structure, all entries having the size of the image. The solution for the  $\mathbf{x}$ - and  $\mathbf{t}$ - update is shown in detail by Shirai and Okuda [42]. Note, however, that both  $\mathbf{x}$ - and  $\mathbf{t}$ -updates minimize the same Lagrangian, but with respect

to different variables. The minimization of  $\mathbf{z}_{1,i}$  and  $\mathbf{z}_{2,j}$  can be done via the ‘‘block soft threshold operator’’  $\mathcal{S}_\gamma$  [42], which is given as follows [42]:

$$\mathcal{S}_\gamma(\mathbf{v}) = \max \left\{ \left( 1 - \frac{\gamma}{\|\mathbf{v}\|_2} \right) \mathbf{v}, 0 \right\}, \tag{37}$$

such that:

$$\mathbf{z}_{1,i}^{k+1} = \mathcal{S}_{\lambda_0/\rho}(\mathbf{D}\mathbf{x}^{k+1} - \mathbf{t}^{k+1} + \mathbf{u}_1^k), \tag{38}$$

$$\mathbf{z}_{2,j}^{k+1} = \mathcal{S}_{\lambda_1/\eta}(\mathbf{G}\mathbf{t}^{k+1} + \mathbf{u}_2^k), \tag{39}$$

with  $i$  and  $j$  denoting the respective direction of the derivative, which in 2D is  $i \in \{h, v\}$  for (38) and  $j \in \{h, d, v\}$  for (39). This operator is applied pixel-wise to the image.

### 2.2 Poisson noise ADMM with TGV

Incorporating the log-likelihood functional (7) directly into ADMM leads to problems of finding a straightforward solution for the  $\mathbf{x}$ - and  $\mathbf{t}$ -update. Thus, it is usually used as a penalizer functional  $g(\mathbf{z})$ , as these problems do not occur then. For data corrupted with Poisson noise, the minimization states as follows:

$$\begin{aligned} & \arg \min_{\mathbf{x}, \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2} G\mathcal{P}(\mathbf{z}_0) + \lambda_0 \|\mathbf{z}_1\|_F + \lambda_1 \|\mathbf{z}_2\|_F \\ & \text{subject to } \mathbf{z}_0 = \mathbf{\Omega}\mathbf{x}, \mathbf{z}_1 = \mathbf{D}\mathbf{x} - \mathbf{t}, \mathbf{z}_2 = \mathbf{G}\mathbf{t} \end{aligned} \tag{40}$$

with the corresponding augmented Lagrangian:

$$\begin{aligned} \mathcal{L}_{\varphi, \rho, \eta}(\mathbf{x}, \mathbf{t}, \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2) = & G\mathcal{P}(\mathbf{z}_0) + \varphi/2 \|\mathbf{\Omega}\mathbf{x} - \mathbf{z}_0 + \mathbf{u}_0\|_F^2 - \varphi/2 \|\mathbf{u}_0\|_F^2 \\ & + \lambda_0 \|\mathbf{z}_1\|_F + \rho/2 \|\mathbf{D}\mathbf{x} - \mathbf{t} - \mathbf{z}_1 + \mathbf{u}_1\|_F^2 - \rho/2 \|\mathbf{u}_1\|_F^2 \\ & + \lambda_1 \|\mathbf{z}_2\|_F + \eta/2 \|\mathbf{G}\mathbf{t} - \mathbf{z}_2 + \mathbf{u}_2\|_F^2 - \eta/2 \|\mathbf{u}_2\|_F^2. \end{aligned} \tag{41}$$

As initial values for the algorithm, the following are chosen:  $\mathbf{x}^0 = \mathbf{0}, \mathbf{t}_i^0 = \mathbf{0}, \mathbf{z}_0^0 = \boldsymbol{\xi}, \mathbf{z}_{1,i}^0 = \mathbf{0}, \mathbf{z}_{2,j}^0 = \mathbf{0}, \mathbf{u}_0^0 = \mathbf{0}, \mathbf{u}_{1,i}^0 = \mathbf{0}$  and  $\mathbf{u}_{2,j}^0 = \mathbf{0}$ . With this, the iteration steps then are given as:

$$\begin{aligned} \left\{ \mathbf{x}^{k+1}, \mathbf{t}^{k+1} \right\} &= \arg \min_{\mathbf{x}, \mathbf{t}} \mathcal{L}_{\varphi, \rho, \eta}(\mathbf{x}, \mathbf{t}, \mathbf{z}_0^k, \mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{u}_0^k, \mathbf{u}_1^k, \mathbf{u}_2^k), \\ \mathbf{z}_0^{k+1} &= \arg \min_{\mathbf{z}_0} \mathcal{L}_\varphi(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_0, \mathbf{u}_0^k), \\ \mathbf{z}_{1,i}^{k+1} &= \arg \min_{\mathbf{z}_{1,i}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_1, \mathbf{u}_1^k), \\ \mathbf{z}_{2,j}^{k+1} &= \arg \min_{\mathbf{z}_{2,j}} \mathcal{L}_\eta(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_2, \mathbf{u}_2^k), \\ \mathbf{u}_0^{k+1} &= \mathbf{u}_0^k + \left( \mathbf{\Omega}\mathbf{x}^{k+1} - \mathbf{z}_0^{k+1} \right), \end{aligned} \tag{42}$$

$$\begin{aligned} \mathbf{u}_{1,i}^{k+1} &= \mathbf{u}_{1,i}^k + \left( [\mathbf{D}\mathbf{x}^{k+1}]_i - \mathbf{t}_i^{k+1} - \mathbf{z}_{1,i}^{k+1} \right) , \\ \mathbf{u}_{2,j}^{k+1} &= \mathbf{u}_{2,j}^k + \left( [\mathbf{G}\mathbf{t}^{k+1}]_j - \mathbf{z}_{2,j}^{k+1} \right) , \end{aligned}$$

The resulting Lagrangian can be solved in the same manner as the Gaussian noise ADMM as shown by Shirai and Okuda [42]. However, despite the slight changes in the  $\mathbf{x}$ - and  $\mathbf{t}$ -update, the only difference in the iteration pattern is the additional  $\mathbf{z}_0$  and  $\mathbf{u}_0$ -update step, together with the introduction of a new penalty parameter  $\varphi$ . The minimization with respect to  $\mathbf{z}_0$  is given by the Poisson noise operator, which equates to the following [34]:

$$\mathbf{z}_0^{k+1} = \text{prox}_{\mathcal{P}_\varphi} \left( \boldsymbol{\Omega}\mathbf{x}^{k+1} + \mathbf{u}_0^k \right) , \tag{43}$$

with the operator being defined as follows [34]:

$$\text{prox}_{\mathcal{P}_\varphi}(\mathbf{v}) = -\frac{1 - \varphi\mathbf{v}}{2\varphi} + \sqrt{\left(\frac{1 - \varphi\mathbf{v}}{2\varphi}\right)^2 + \frac{\boldsymbol{\xi}}{\varphi}} . \tag{44}$$

Again, this operator is applied pixel-wise. All other  $\mathbf{z}$ -updates are carried out as in the Gaussian ADMM.

### 3 ADMM-TGV with residual balancing

When utilizing the residual balancing for ADMM-TGV algorithms, as described in Sect. 1.4,  $\rho$ ,  $\eta$ , and  $\varphi$  are to be balanced here. So, the overall structure can be divided into two parts for the Gaussian algorithm—one for the  $\rho$ -balancing and one for the  $\eta$ -balancing. Further, a third part for the Poisson algorithm is needed—the  $\varphi$ -balancing.

#### 3.1 Primal feasibility conditions of ADMM-TGV

Since  $\rho$ ,  $\eta$ , and  $\varphi$  are scalars, acting on all pixels equally, we need to find global residuals, depending on all pixels. Following the pattern of Sect. 1.4 by deriving the augmented Lagrangian for Gaussian (see (34)) or Poisson noise (41) with respect to the unscaled dual variables  $\mathbf{y}_{1,i} = \rho\mathbf{u}_{1,i}$  and  $\mathbf{y}_{2,j} = \rho\mathbf{u}_{2,j}$  (and additionally  $\mathbf{y}_0 = \rho\mathbf{u}_0$  for Poisson) for all respective directions leads to the primal feasibility conditions with their respective relative residuals:

$$R_{rel,\rho}^{k+1} = \frac{\|\mathbf{R}_\rho^{k+1}\|_F}{N(\mathbf{R}_\rho^{k+1})} = \frac{\|\mathbf{D}\mathbf{x}^{k+1} - \mathbf{t}^{k+1} - \mathbf{z}_1^{k+1}\|_F}{\max\left\{\|\mathbf{D}\mathbf{x}^{k+1} - \mathbf{t}^{k+1}\|_F, \|\mathbf{z}_1^{k+1}\|_F\right\}} , \tag{45}$$

$$R_{rel,\eta}^{k+1} = \frac{\|\mathbf{R}_\eta^{k+1}\|_F}{N(\mathbf{R}_\eta^{k+1})} = \frac{\|\mathbf{G}\mathbf{t}^{k+1} - \mathbf{z}_2^{k+1}\|_F}{\max\left\{\|\mathbf{G}\mathbf{t}^{k+1}\|_F, \|\mathbf{z}_2^{k+1}\|_F\right\}} , \tag{46}$$

with the standard primal residual  $\|\mathbf{R}_\rho^{k+1}\|_F$  in the numerator and the normalization for the scaled residual  $N(\mathbf{R}_\rho^{k+1})$  in the denominator. Here,  $\mathbf{D}\mathbf{x}^{k+1} - \mathbf{t}^{k+1} = \mathbf{s}^{k+1}$  form

a single norm in the normalization (45) instead of appearing as separated norms like (23) suggests, because both are coupled via (31).

For the Poisson TGV, a primal residual for the new penalty parameter  $\varphi$  can be found as follows:

$$R_{rel,\varphi}^{k+1} = \frac{\|\mathbf{R}_\varphi^{k+1}\|_F}{N(\mathbf{R}_\varphi^{k+1})} = \frac{\|\Omega\mathbf{x}^{k+1} - \mathbf{z}_0^{k+1}\|_F}{\max\{\|\Omega\mathbf{x}^{k+1}\|_F^2, \|\mathbf{z}_0^{k+1}\|_F^2\}}. \tag{47}$$

### 3.2 Dual feasibility conditions of ADMM-TGV

Since the updates happen jointly in (35) and (42),  $\mathbf{x}^k$  and all  $\mathbf{t}_i^k$  are not updated yet, regardless which variable we consider. To find the whole set of dual residuals, the augmented Lagrangian has to be derived with respect to  $\mathbf{x}$  and to all  $\mathbf{t}_i$ . This leads to the equations:

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} \mathcal{L}_{\rho,\eta}(\mathbf{x}, \mathbf{t}^k, \mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{u}_1^k, \mathbf{u}_2^k) \quad , \tag{48}$$

$$\mathbf{t}_i^{k+1} = \arg \min_{\mathbf{t}_i} \mathcal{L}_{\rho,\eta}(\mathbf{x}^k, \mathbf{t}_i, \mathbf{t}_l^k, \mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{u}_1^k, \mathbf{u}_2^k), \tag{49}$$

with  $\mathbf{t}_l$  denoting all other vector elements of  $\mathbf{t}$  excluding the  $i$ -th element (for images  $l \in \{h, v \mid l \neq i\}$ ) as both  $\mathbf{x}$  and  $\mathbf{t}_i$  rely on the former iterates of the respective other variables.

Deriving the augmented Lagrangian with respect to  $\mathbf{x}$  leads to the first dual feasibility condition:

$$\begin{aligned} 0 &= \partial_{\mathbf{x}} f(\mathbf{x}) + \rho^k \mathbf{D}^T (\mathbf{D}\mathbf{x}^{k+1} - \mathbf{t}^k - \mathbf{z}_1^k + \mathbf{u}_1^k) \\ &= \partial_{\mathbf{x}} f(\mathbf{x}) + \rho^k \mathbf{D}^T (\mathbf{u}_1^{k+1} + \mathbf{t}^{k+1} + \mathbf{z}_1^{k+1} - \mathbf{t}^k - \mathbf{z}_1^k) \\ &\Leftrightarrow -\rho^k \mathbf{D}^T (\mathbf{t}^{k+1} + \mathbf{z}_1^{k+1} - \mathbf{t}^k - \mathbf{z}_1^k) = 1/\sigma^2 \Omega^T (\Omega\mathbf{x} - \xi) + \rho^{k+1} \mathbf{D}^T \mathbf{u}_1^{k+1}, \end{aligned} \tag{50}$$

which can be seen as the dual residual, with  $f(\mathbf{x}) = 1/2\sigma^2\|\Omega\mathbf{x} - \xi\|_F^2$  (see (34)) for the Gaussian algorithm. The corresponding relative dual residual is given as follows:

$$S_{rel,\rho,x}^{k+1} = \frac{\|\mathbf{S}_{\rho,x}^{k+1}\|_F}{N(\mathbf{S}_{\rho,x}^{k+1})} = \frac{\|-\rho^k [\mathbf{D}^T (\mathbf{t}^{k+1} - \mathbf{t}^k + \mathbf{z}_1^{k+1} - \mathbf{z}_1^k)]\|_F}{\max\{\|1/\sigma^2 \Omega^T (\Omega\mathbf{x} - \xi)\|_F, \|\rho^k \mathbf{D}^T (\mathbf{u}_1^k + \mathbf{R}_1^{k+1})\|_F\}}, \tag{51}$$

with the standard dual residual  $\|\mathbf{S}_{\rho,x}^{k+1}\|$  in the numerator and the normalization for the scaled residual  $N(\mathbf{S}_{\rho,x}^{k+1})$  in the denominator.

For the Poisson algorithm, however, the dual feasibility condition equates to the following:

$$-\varphi^k \boldsymbol{\Omega}^T (\mathbf{z}_0^{k+1} - \mathbf{z}_0^k) - \rho^k \mathbf{D}^T (\mathbf{t}^{k+1} + \mathbf{z}_1^{k+1} - \mathbf{t}^k - \mathbf{z}_1^k) = \varphi^{k+1} \boldsymbol{\Omega}^T \mathbf{u}_0^{k+1} + \rho^{k+1} \mathbf{D}^T \mathbf{u}_1^{k+1}, \tag{52}$$

with  $f(\mathbf{x}) = 0$  (see (41)). Since all residuals must vanish eventually, we can rearrange the dual residual into a residual for the balancing of  $\rho$  and for  $\varphi$ :

$$S_{rel,\rho,x}^{k+1} = \frac{\|\mathbf{S}_{\rho,x}^{k+1}\|_F}{N(\mathbf{S}_{\rho,x}^{k+1})} = \frac{\|\mathbf{S}_{\rho,x}^{k+1}\|_F}{\max\{\|\mathbf{N}^*(\mathbf{S}_{\rho,x}^{k+1})\|_F, \|\mathbf{N}^*(\mathbf{S}_{\rho,x}^{k+1}) - \mathbf{S}_{\rho,x}^{k+1}\|_F\}}, \tag{53}$$

$$S_{rel,\varphi,x}^{k+1} = \frac{\|\mathbf{S}_{\varphi,x}^{k+1}\|_F}{N(\mathbf{S}_{\varphi,x}^{k+1})} = \frac{\|\mathbf{S}_{\varphi,x}^{k+1}\|_F}{\max\{\|\mathbf{N}^*(\mathbf{S}_{\varphi,x}^{k+1})\|_F, \|\mathbf{N}^*(\mathbf{S}_{\varphi,x}^{k+1}) - \mathbf{S}_{\varphi,x}^{k+1}\|_F\}}, \tag{54}$$

with:

$$\begin{aligned} \mathbf{S}_{\rho,x}^{k+1} &= -\rho^k \left[ \mathbf{D}^T (\mathbf{t}^{k+1} - \mathbf{t}^k + \mathbf{z}_1^{k+1} - \mathbf{z}_1^k) \right], \\ \mathbf{N}^*(\mathbf{S}_{\rho,x}^{k+1}) &= \rho^k \mathbf{D}^T (\mathbf{u}_1^k + \mathbf{R}_1^{k+1}), \\ \mathbf{S}_{\varphi,x}^{k+1} &= -\varphi^k \boldsymbol{\Omega}^T (\mathbf{z}_0^{k+1} - \mathbf{z}_0^k), \\ \mathbf{N}^*(\mathbf{S}_{\varphi,x}^{k+1}) &= \varphi^k \boldsymbol{\Omega}^T (\mathbf{u}_0^k + \mathbf{R}_0^{k+1}). \end{aligned}$$

For both Poisson and Gaussian, (51) and (53) show that a change between iterations in  $\mathbf{z}_{1,i}$  can be induced by a change in  $\mathbf{t}$ , but then does not change the residual for the  $\mathbf{x}$ -update.

Deriving the augmented Lagrangian ((34) or (41)) with respect to the dual variables  $\mathbf{t}_i$  with  $n$ -dimensions leads to the following:

$$\begin{aligned} 0 &= -\rho^k (\mathbf{D}_i \mathbf{x}^k - \mathbf{t}_i^{k+1} - \mathbf{z}_{1,i}^{k+1} + \mathbf{u}_{1,i}^k) \\ &+ \eta^k \left( \begin{bmatrix} \mathbf{G}^T \\ \mathbf{G}^T \end{bmatrix}_i \right) \left[ \left( \begin{bmatrix} \mathbf{G}^T \\ \mathbf{G}^T \end{bmatrix}_i^T \right) \begin{pmatrix} \mathbf{t}_i^{k+1} \\ \mathbf{t}_i^k \end{pmatrix} \right] + \eta^k \mathbf{D}_i (\mathbf{u}_{2,i}^k - \mathbf{z}_{2,i}^k) \\ &+ \eta^k \sum_l \mathbf{D}_l (\mathbf{u}_{2,il}^k - \mathbf{z}_{2,il}^k), \end{aligned} \tag{55}$$

with  $i$  being the direction of the derivative (or vector row) and  $(il)$  denoting all the diagonal vector elements for  $i$ , here for images  $(il) \in \{hv, vh\} = \{d\}$ . Solving for the relative dual residuals connected to both  $\rho$  and  $\eta$  leads to the following:

$$S_{rel,\rho,t_i}^{k+1} = \frac{\|\mathbf{S}_{\rho,t_i}^{k+1}\|_F}{N(\mathbf{S}_{\rho,t_i}^{k+1})} = \frac{\|\mathbf{S}_{\rho,t_i}^{k+1}\|_F}{\max\{\|\mathbf{N}^*(\mathbf{S}_{\rho,t_i}^{k+1})\|_F, \|\mathbf{N}^*(\mathbf{S}_{\rho,t_i}^{k+1}) - \mathbf{S}_{\rho,t_i}^{k+1}\|_F\}}, \tag{56}$$

$$S_{rel,\eta,t_i}^{k+1} = \frac{\|S_{\eta,t_i}^{k+1}\|_F}{N(S_{\eta,t_i}^{k+1})} = \frac{\|S_{\eta,t_i}^{k+1}\|_F}{\max\{\|N^*(S_{\eta,t_i}^{k+1})\|_F, \|N^*(S_{\rho,t_i}^{k+1}) - S_{\rho,t_i}^{k+1}\|_F\}}, \quad (57)$$

with:

$$\begin{aligned} S_{\rho,t_i}^{k+1} &= \rho^k \left[ D_i (\mathbf{x}^k - \mathbf{x}^{k+1}) + \mathbf{z}_{1,i}^{k+1} - \mathbf{z}_{1,i}^k \right], \\ N^*(S_{\rho,t_i}^{k+1}) &= \rho^k \left[ \mathbf{u}_{1,i}^k + \mathbf{R}_{1,i}^{k+1} \right], \\ S_{\eta,t_i}^{k+1} &= -\eta^k \left[ D_i (\mathbf{z}_{2,i}^{k+1} - \mathbf{z}_{2,i}^k) + \sum_l D_l (D_i^T (\mathbf{t}_i^k - \mathbf{t}_l^{k+1}) + \mathbf{z}_{2,il}^{k+1} - \mathbf{z}_{2,il}^k) \right], \\ N^*(S_{\eta,t_i}^{k+1}) &= \eta^k \left[ D_i (\mathbf{u}_{2,i}^k + \mathbf{R}_{2,i}^{k+1}) + \sum_l D_l (\mathbf{u}_{2,il}^k + \mathbf{R}_{2,il}^{k+1}) \right]. \end{aligned}$$

These equations show that a change in  $\mathbf{z}_{1,i}$  in between iterations might be induced by a change in  $\mathbf{x}$ , which then also does not appear in the residual for the  $\mathbf{t}_i$  filter mask update. The same holds for a change in the diagonal  $\mathbf{z}_{il}$  elements, which can occur due to a change in the other  $\mathbf{t}_l$  filter masks.

Since  $\mathbf{x}$  and  $\mathbf{t}_i$  can be represented as a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{t} \end{pmatrix}$ , the same holds for the dual residuals.

Thus, applying the Frobenius norm to form a global relative residual for the  $\rho$ ,  $\eta$ , and  $\varphi$  balancing seems logical:

$$S_{rel,\rho}^{k+1} = \sqrt{\frac{\|S_{\rho,x}^{k+1}\|_F^2 + \sum_i \|S_{\rho,t_i}^{k+1}\|_F^2}{N(S_{\rho,x}^{k+1})^2 + \sum_i N(S_{\rho,t_i}^{k+1})^2}}, \quad (58)$$

$$S_{rel,\eta}^{k+1} = \sqrt{\frac{\sum_i \|S_{\eta,t_i}^{k+1}\|_F^2}{\sum_i N(S_{\eta,t_i}^{k+1})^2}}, \quad (59)$$

$$S_{rel,\varphi}^{k+1} = \frac{\|S_{\varphi,x}^{k+1}\|_F}{N(S_{\varphi,x}^{k+1})}. \quad (60)$$

The dual residuals arising from the derivation with respect to  $\mathbf{t}$  stay identical to the Poisson ADMM-TGV.

These primal and dual residuals are used for balancing  $\rho$  and  $\eta$  in the following according to (28). The residual balanced algorithms are hereafter called ADMM-RBTGV for Gauss or Poisson noise.

### 3.3 ADMM for Gaussian and Poisson noise with RBTGV

Implementing the residual balanced algorithm for Gaussian noise gives the following algorithm, with starting conditions  $\mathbf{x}^0 = 0$ ,  $\mathbf{t}_i^0 = 0$ ,  $\mathbf{z}_{1,i}^0 = 0$ ,  $\mathbf{z}_{2,j}^0 = 0$ ,  $\mathbf{u}_{1,i}^0 = 0$ ,

$\mathbf{u}_{2,j}^0 = \mathbb{0}$  and finally  $\rho^0 = 1, \eta^0 = 1$ , leading to the following:

$$\{\mathbf{x}^{k+1}, \mathbf{t}^{k+1}\} = \arg \min_{\mathbf{x}, \mathbf{t}} \mathcal{L}_{\rho^k, \eta^k}(\mathbf{x}, \mathbf{t}, \mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{u}_1^k, \mathbf{u}_2^k) \quad , \quad (\text{after [42]})$$

$$\mathbf{z}_{1,i}^{k+1} = \arg \min_{\mathbf{z}_{1,i}} \mathcal{L}_{\rho^k}(\mathbf{x}^{k+1}, \mathbf{t}_i^{k+1}, \mathbf{z}_{1,i}, \mathbf{u}_{1,i}^k) \quad , \quad (\text{after (38)})$$

$$\mathbf{z}_{2,j}^{k+1} = \arg \min_{\mathbf{z}_{2,j}} \mathcal{L}_{\eta^k}(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_{2,j}, \mathbf{u}_{2,j}^k) \quad , \quad (\text{after (39)})$$

if  $k \in \text{Val}$

$$\tau_\rho^{k+1} \leftarrow \tau_\rho^k, \quad \tau_\eta^{k+1} \leftarrow \tau_\eta^k \quad , \quad (\text{after (28)}) \quad (61)$$

$$\rho^{k+1} \leftarrow \rho^k, \quad \eta^{k+1} \leftarrow \eta^k \quad , \quad (\text{after (29)})$$

else

$$\tau_\rho^{k+1} = 1, \quad \tau_\eta^{k+1} = 1 \quad ,$$

end

$$\mathbf{u}_{1,i}^{k+1} = \frac{\mathbf{u}_{1,i}^k + ([\mathbf{D}\mathbf{x}^{k+1}]_i - \mathbf{t}_i^{k+1} - \mathbf{z}_{1,i}^{k+1})}{\tau_\rho^{k+1}} \quad ,$$

$$\mathbf{u}_{2,j}^{k+1} = \frac{\mathbf{u}_{2,j}^k + ([\mathbf{G}\mathbf{t}^{k+1}]_j - \mathbf{z}_{2,j}^{k+1})}{\tau_\eta^{k+1}} \quad ,$$

with  $\text{Val}$  being a series of exponentially growing values, such that the residual balancing occurs less and less often. The initial values are the same as in the algorithm for the standard Gaussian ADMM with TGV (35).

The residual balanced algorithm for Poisson noise, with the starting conditions of the Gaussian algorithm plus  $\mathbf{z}_{1,i}^0 = \boldsymbol{\xi}$  and  $\mathbf{u}_0^0 = \mathbb{0}$ , is given as follows:

$$\{\mathbf{x}^{k+1}, \mathbf{t}^{k+1}\} = \arg \min_{\mathbf{x}, \mathbf{t}} \mathcal{L}_{\varphi^k, \rho^k, \eta^k}(\mathbf{x}, \mathbf{t}, \mathbf{z}_0^k, \mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{u}_0^k, \mathbf{u}_1^k, \mathbf{u}_2^k) \quad , \quad (\text{after [42]})$$

$$\mathbf{z}_0^{k+1} = \arg \min_{\mathbf{z}_0} \mathcal{L}_{\varphi^k}(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_0, \mathbf{u}_0^k) \quad , \quad (\text{after (47)})$$

$$\mathbf{z}_{1,i}^{k+1} = \arg \min_{\mathbf{z}_{1,i}} \mathcal{L}_{\rho^k}(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_{1,i}, \mathbf{u}_1^k) \quad , \quad (\text{after (38)})$$

$$\mathbf{z}_{2,j}^{k+1} = \arg \min_{\mathbf{z}_{2,j}} \mathcal{L}_{\eta^k}(\mathbf{x}^{k+1}, \mathbf{t}^{k+1}, \mathbf{z}_{2,j}, \mathbf{u}_2^k) \quad , \quad (\text{after (39)})$$

if  $k \in \text{Val}$

$$\tau_\rho^{k+1} \leftarrow \tau_\rho^k, \quad \tau_\eta^{k+1} \leftarrow \tau_\eta^k, \quad \tau_\varphi^{k+1} \leftarrow \tau_\varphi^k \quad , \quad (\text{after (28)}) \quad (62)$$

$$\rho^{k+1} \leftarrow \rho^k, \quad \eta^{k+1} \leftarrow \eta^k, \quad \varphi^{k+1} \leftarrow \varphi^k \quad , \quad (\text{after (29)})$$

else

$$\tau_\rho^{k+1} = 1, \quad \tau_\eta^{k+1} = 1, \quad \tau_\varphi^{k+1} = 1 \quad ,$$

end

$$\mathbf{u}_0^{k+1} = \frac{\mathbf{u}_0^k + (\boldsymbol{\Omega}\mathbf{x}^{k+1} - \mathbf{z}_0^{k+1})}{\tau_\varphi^{k+1}} \quad ,$$



$$\begin{aligned} \mathbf{u}_{1,i}^{k+1} &= \frac{\mathbf{u}_{1,i}^k + ([\mathbf{D}\mathbf{x}^{k+1}]_i - \mathbf{t}_i^{k+1} - \mathbf{z}_{1,i}^{k+1})}{\tau_\rho^{k+1}}, \\ \mathbf{u}_{2,j}^{k+1} &= \frac{\mathbf{u}_{2,j}^k + ([\mathbf{G}\mathbf{t}^{k+1}]_j - \mathbf{z}_{2,j}^{k+1})}{\tau_\eta^{k+1}}. \end{aligned}$$

### 4 Evaluation of the algorithms

In this section, the algorithms are tested on images for denoising and deconvolution problems to demonstrate the success of residual balancing and to show its impact on the convergence of the respective algorithm.

An image is superimposed with noise, such that it is easy to compare the restoration with the original image and show which parameter sets work best. For TGV, four parameters have to be well chosen to achieve optimality,  $\rho$  and  $\eta$ ,  $\lambda_0$  and  $\lambda_1$ ; for Poisson, this list extends by an additional penalty parameter  $\varphi$ .

For evaluating the algorithm, a quality measure is defined. For the quality of the restoration, the mean squared error (MSE) after iteration  $k$

$$\text{MSE}^k = \frac{1}{M \cdot N} \sum_{m,n} (\mathbf{x}_{i,j}^k - \xi_{i,j}^{Orig})^2, \tag{63}$$

is a commonly used criterion, with the original image  $\xi^{Orig}$  known and the current iterate image  $\mathbf{x}^k$ . In order to make optimal regions for the parameter choices visible, initially,  $\rho = \eta (= \varphi$  for Poisson) and  $\lambda_0 = \lambda_1$  are chosen over orders of magnitude from  $10^{-5} \leq \rho \leq 10^5$  and  $10^{-5}/\omega\sigma \leq \lambda_0 \leq 10^5/\omega\sigma$ , where  $\sigma$  is the standard deviation of the inherent noise in the image and  $\omega$  is an attenuation coefficient depending on the deconvolution kernel.

Looking at the total generalized variation ADMM approach for Gaussian noise (33) and setting  $a \cdot \lambda = \lambda_0$ ,  $b \cdot \lambda = \lambda_1$ , with  $\mathbf{\Omega} = \mathbb{1}$ ,

$$\begin{aligned} \arg \min_{\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2} & \quad 1/2\sigma^2 \|\mathbf{x} - \xi\|_F^2 + \lambda (a \cdot \|\mathbf{z}_1\|_F + b \cdot \|\mathbf{z}_2\|_F) \\ \text{subject to} & \quad \mathbf{z}_1 = \mathbf{D}\mathbf{x} - \mathbf{t}, \mathbf{z}_2 = \mathbf{G}\mathbf{t} \end{aligned}, \tag{64}$$

the following can be seen: If the true, noise-free image  $\mathbf{x}^*$  could be found, the first term of the minimization would equal:

$$\frac{1}{2\sigma^2} \|\mathbf{x}^* - \xi\|_F^2 = \sum_{n,m} \frac{1}{2} \iff \|\mathbf{x}^* - \xi\|_F^2 = \sum_{n,m} \sigma^2, \tag{65}$$

by definition of the standard deviation  $\sigma$ , as  $\sigma$  characterizes the mean noise level of all

contributing pixels. In a perfect setting with perfect noise detection, the TGV prior would be that exact difference:

$$\lambda \cdot (a \cdot \|\mathbf{z}_1\|_F + b \cdot \|\mathbf{z}_2\|_F) = \lambda \cdot \|\mathbf{x}^* - \boldsymbol{\xi}\|_F = \left( \sum_{n,m} \lambda^2 \cdot \sigma^2 \right)^{1/2}. \tag{66}$$

Since we have perfect noise estimation in this example, the noise level would be irrelevant for the noise detection and for the image. Putting (65) and (66) into the minimization (64) shows that the minimum only depends on the total amount of pixels, if  $\lambda = 1/\sigma$  is chosen as an optimal regularization parameter:

$$\arg \min_{\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2} \sum_{n,m} \frac{1}{2} + \left( \sum_{n,m} \lambda^2 \cdot \sigma^2 \right)^{1/2}. \tag{67}$$

But as the image can only be separated into piece-wise flat or sloped regions either connected to  $\|\mathbf{z}_1\|_F$  or  $\|\mathbf{z}_2\|_F$ ,  $a$  and  $b$  must add to 1 to comprise the whole image. However, as noise cannot be predicted perfectly, the optimal  $\lambda$  deviates according to the quality and applicability of the prior. In case of a bad prior, distortions are induced and may at some point outweigh the benefits to the restoration process. Conversely, this would reduce the optimal regularization parameter compared to what would be the case with a good prior. However, as most images comprise piece-wise flat or sloped regions, it is reasonable to assume that TGV is close to approaching the optimal regularization parameter of  $\lambda = 1/\sigma$ . If the standard deviation of the noise  $\sigma$  is sufficiently known, optimality mainly depends on the choice of  $a$  and  $b$  weighting  $\|\mathbf{z}_1\|_F$  and  $\|\mathbf{z}_2\|_F$ , respectively.

The same argumentation also holds for Poisson noise, as it approaches Gaussian noise for the increasing number of counts. However, for deconvolution, the  $\mathbf{z}_1$  and  $\mathbf{z}_2$  are applied to the deconvolved estimate of the image  $\mathbf{x}$  and not to the convolved estimate  $\boldsymbol{\Omega}\mathbf{x}$  (see (35) and (42)). The noise is thus attenuated by a factor  $\omega$  which depends on the convolution kernel the image is convolved with. The wider the kernel  $\boldsymbol{\Omega}$  for deconvolution, the more of the noise gets drawn into a single pixel. By normalizing the kernel to its maximum peak, we can find the contributions of the other pixel noises to the attenuated noise. Since adding noises corresponds to adding their variances, this leads to the following:

$$\omega^2 = \sum_{i,j} \frac{\boldsymbol{\Omega}_{i,j}}{\max \{ \boldsymbol{\Omega}_{i,j} \}}. \tag{68}$$

So, the overall scaling needs to be  $1/\omega\sigma$ . For denoising, the attenuation is  $\omega^2 = 1$ , since  $\boldsymbol{\Omega}$  corresponds to a delta peak function.

To make regions visible where the overall quality improves the MSE of a given iteration,  $k$  can be divided by the initial MSE<sup>0</sup>:

$$\text{NormMSE}^k = \frac{\text{MSE}^k}{\text{MSE}^0}, \tag{69}$$

showing how much the image improves relatively.

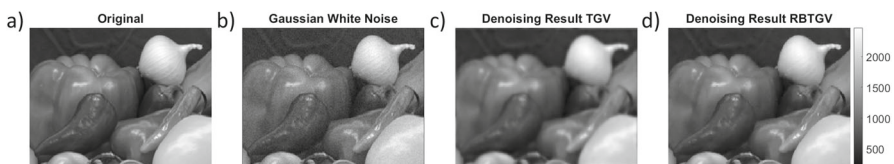
### 4.1 Denoising of images with Gaussian noise

The first original image can be found in Fig. 1a. In Fig. 1b, Gaussian noise of  $\sigma = 50$  was added to the image to achieve a signal-to-noise ratio of  $\text{SNR} \approx 20$ . To visualize the effect of residual balancing, a suboptimal choice of the penalty parameters  $\rho$  and  $\eta$  was done, while the regularization parameters  $\lambda_0$  and  $\lambda_1$  were kept close to the optimum. In Fig. 1 c and d, the denoised images for the ADMM-TGV and ADMM-RBTGV algorithms after 1000 iterations are compared.

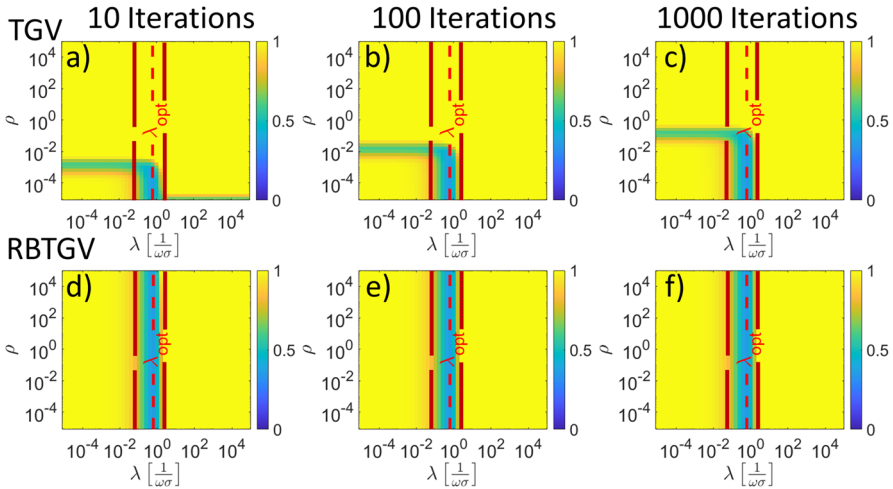
The difference between both denoised images Fig. 1 c and d is rather easy to spot, as c looks blurry compared to d. The suboptimal choice of  $\rho$  and  $\eta$  diminishes the progress of the denoising. However, as residual balancing equalizes the denoising result over all penalty parameters, making the result independent of the initial choice, the progress of the algorithm is further advanced, and thus, the quality of the reconstruction is significantly improved.

To show that penalty parameters  $\rho$  and  $\eta$  for residual balancing can in fact be arbitrarily chosen, Fig. 2 displays the course of thousand iterations for a wide range of regularization parameters. The NormMSE (see (69)) is displayed such that it shows parameter sets improving the image quality in blue and the rest in yellow. This depiction allows to quickly get an insight into the relevant range of the parameters, that an operator would preferably choose from.

It can be seen that the original ADMM-TGV algorithm in the upper row of Fig. 2 strongly depends on the initial choice of  $\rho$  and  $\eta$  as the blue-colored optimal region slowly shifts to higher  $\rho$  and  $\eta$  in the course of iterations. With a higher number of iterations, the NormMSE-map of the original ADMM-TGV forms an inverted L-shaped optimal region that grows in  $\rho$ -direction with increasing the number of iterations. Thus, the optimality map can be divided into three regions: the first region at low values of  $\lambda$  bordered by the first red line, in which the algorithm does not improve the image quality significantly, because the noise is estimated too low, leading to under-smoothing; the second, inside the red boundaries, where the algorithm converges to a better-than-before solution around the optimum; and a third region in which the algorithm decreases image quality due to over-smoothing, eventually. In the left outer regions at some point in the iterations, an optimum is reached for some  $\rho$  at a given



**Fig. 1** a Test image “onion” from MATLAB. The pixel intensities were multiplied by a factor of 10. b Image with additional Gaussian white noise with a standard deviation of  $\sigma = 50$  and the attenuation  $\omega = 1$ . c The result of the original ADMM-TGV (see (35)) and d the result of our ADMM-RBTGV algorithm (see (61)), after 1000 iterations and with unfavorable starting values for the penalty parameters  $\rho = \eta = 1$  and optimal regularization parameters  $\lambda_0 = \lambda_1 \approx 0.63/\omega\sigma$  as values, respectively



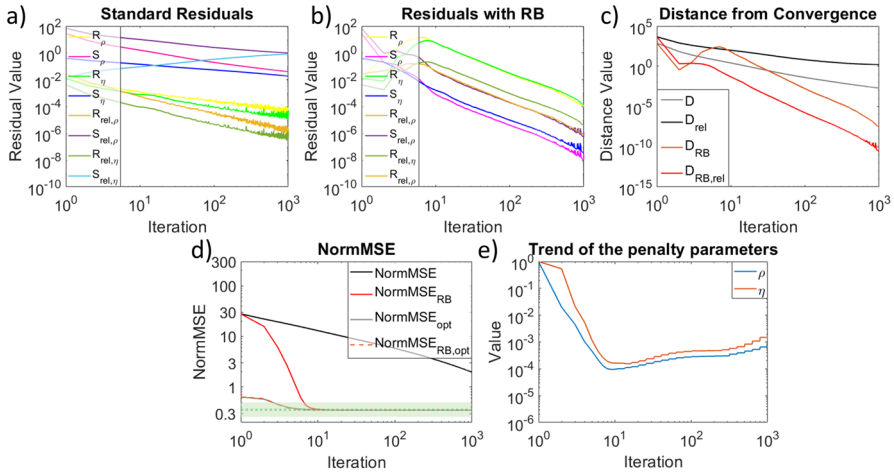
**Fig. 2** NormMSE maps generated from applying both ADMM-TGV and ADMM-RBTGV algorithms on Fig. 1, with the individual pixels showing the denoising results under Gaussian noise of the respective  $(\rho = \eta | \lambda_0 = \lambda_1)$ - pair. In the upper row, the results of the original ADMM-TGV are shown after **a** 10, **b** 100, and **c** 1000 iterations. In the lower row (**d–f**), the results of the residual balanced ADMM-RBTGV approach are displayed at the same iteration stages. The two red lines in every map indicate the region, in which the algorithm eventually improves the image quality compared to the initial noisy image, and the dashed line indicates the  $\lambda_0 = \lambda_1$ - value in units of  $1/\omega\sigma$  at which results are optimal ( $\lambda_{opt} \approx 0.63/\omega\sigma$ )

iterate, but lost due to convergence to another result. It is, however, not predictable for a given image, when this incident occurs and thus recommended to stay in the second, optimal region. For every  $\rho$  in this optimal region, the TGV algorithm converges eventually to the same result for a given  $\lambda$ , but might need many iterations due to a poor choice of  $\rho$ .

In order to investigate convergence for all initial choices of  $\rho = \eta$ , the residual balancing is applied in the second row of Fig. 2. Here, the results at a given  $\lambda_0 = \lambda_1$  column are constant for all starting values  $\rho = \eta$  as early as after ten iterations, indicating that the residual balanced approach does not depend heavily on the initial choice of  $\rho$  and  $\eta$ .

To elaborate further, if a faster convergence of the algorithm compared to the original, the drop rate of the  $R$  and  $S$  residuals connected to  $\rho$  and  $\eta$  are evaluated in Figs. 3 a and b, respectively.

The developments of the  $R$  and  $S$  residuals are shown for the optimal choice of  $\lambda_0 = \lambda_1 \approx 0.63/1/\omega\sigma$  and an unfavorable  $\rho = \eta = 1$  value as found in Fig. 2. By comparing the residuals for the ADMM-TGV algorithm in Fig. 3a and the ADMM-RBTGV algorithm in b, it can be seen that the regular rebalancing of the  $R$  and  $S$  pairs leads to an increased drop rate of all residuals. Combining all residuals to a distance measure  $D^{k+1} = \|\mathbf{R}_\rho^{k+1}\|_F^2 + \|\mathbf{S}_\rho^{k+1}\|_F^2 + \|\mathbf{R}_\eta^{k+1}\|_F^2 + \|\mathbf{S}_\eta^{k+1}\|_F^2$  following the scheme of He et al. [37] leads to a clearer picture. In Fig. 3c, the distance  $D$  of the residual balanced ADMM-RBTGV (RB) can be seen to drop faster than the distance for the original ADMM-TGV. These trends stand out even more, when analyzing relative residuals (rel.) in Fig. 3 a and b as defined in (45) and (46) as well as in (58) and



**Fig. 3** **a** Trend of the  $R$  and  $S$  residuals connected to both  $\rho$  and  $\eta$  belonging to the standard ADMM-TGV algorithm for Gaussian noise (Fig. 2d) and the relative residuals  $R_{rel}$  and  $S_{rel}$  described in Sect. 3.1 and Sect. 3.2. Unfavorable starting values of  $\rho = \eta = 1$  and optimal values for  $\lambda_0 = \lambda_1 = \lambda_{opt} \approx 0.63 \ 1/\omega\sigma$  were chosen to compare with **b** the residuals of the residual balanced ADMM-RBTGV denoted as RB. The optimal values for  $\lambda_0 = \lambda_1$  were found as the column with minimal NormMSE values in Fig. 2d. It can be seen that for the residual balancing, the respective  $R$  and  $S$  residuals for  $\rho$  and  $\eta$  are bound to each other and drop faster than for the original algorithm. **c** Combining all four residuals to a distance measure  $D$  confirms increased drop rates for the residual balancing. **d** The NormMSE shows that the RBTGV algorithm (red) converges faster than the original algorithm (black), as it approaches the optimum zone (green) of the converged result (green dotted line) from early on. The NormMSE values for an optimal choice (denoted as opt) of  $\rho = \eta \approx 6.3 \cdot 10^{-5}$ , however, are faster for both algorithms with the residual balancing (orange dotted line) being equally fast as the original algorithm (gray). **e** Development of  $\rho$  and  $\eta$  in the course of the iterations

(60). Here, the relative  $S_{rel,\eta}$  residual increases for the original algorithm whereas it decreases in the residual balanced.

The residual balancing thus results in a faster convergence, as can be seen in Fig. 3d, where the NormMSE value of the residual balanced ADMM-RBTGV settles close to a value of 0.3 after around ten iterations, whereas the non-balanced ADMM-TGV requires several thousand. To set this into perspective, the NormMSE value for the optimal choice of  $\rho = \eta = 6.3 \cdot 10^{-5}$ , as found as the column with minimum values in Fig. 2, settles after four iterations, with no difference between residual balancing and the original algorithm. Thus, ADMM-RBTGV is beneficial when the optimal  $\rho$  and  $\eta$  values are unknown, which in general is true without excessive testing of both values.

In Fig. 3e, the trends of  $\rho$  and  $\eta$  for the residual balanced algorithm are shown, which indicate that the initial (unfavorable) choice was indeed set too high, as the residual balancing corrects these values significantly to lower values.

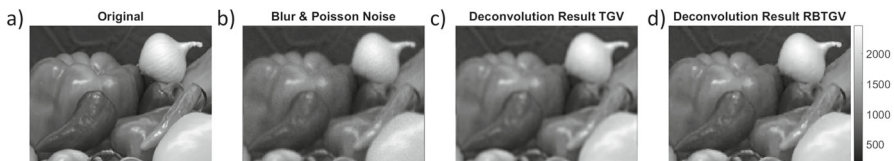
### 4.2 Deconvolution of images with Poisson noise

To further demonstrate the versatile applicability to other types of data reconstruction, in a next example, residual balancing is applied to an image affected by blur and pure

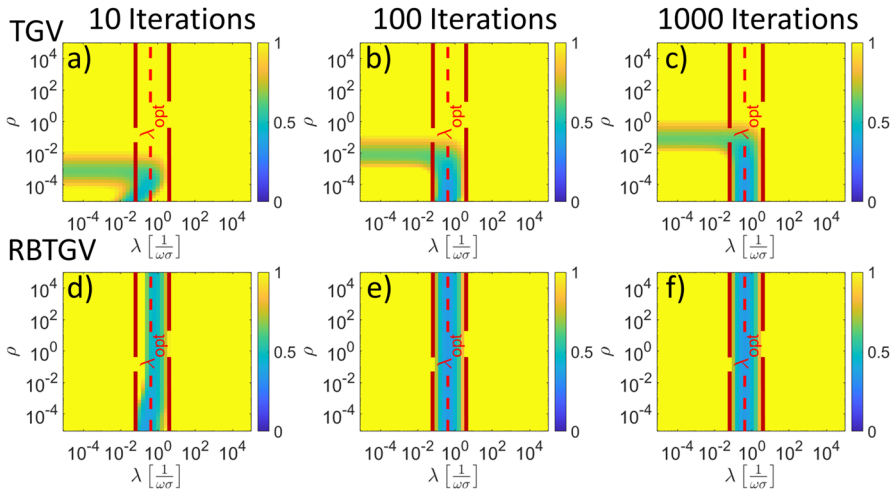
Poisson noise. The original image can be found in Fig. 4a. After convolution with a Gaussian kernel (FWHM = 1 Pixel), Poisson noise ( $\sigma_{mean} \approx 31.7$ ) is added in Fig. 4b. Further, in Fig. 4 c and d, the deconvolved image for the ADMM-TGV and ADMM-RBTGV algorithms after 1000 iterations, respectively, is displayed.

At a glance, the image reconstructed by the ADMM-RBTGV looks sharper compared to the ADMM-TGV-treated image, and further analysis can confirm this: Similar to the Gaussian noise results, regions of optimality for  $\rho = \eta = \varphi$  and  $\lambda_0 = \lambda_1$  can be found via the NormMSE maps as shown in Fig. 5. Again, the convergence of the original algorithm (Fig. 5 a–c) lags behind the ADMM-RBTGV (Fig. 5 d–f) convergence for most  $\rho = \eta = \varphi$  values. Our algorithm shows nearly the same result for all  $\rho$  of a given  $\lambda$  after approximately ten iterations (Fig. 5e), indicating that the residual balancing is applicable here, too. But the convergence is slightly slower than in the Gaussian example, as an additional penalizer for the Poisson noise must be balanced in.

Figure 6a indicates that the residual for  $\rho$ ,  $\eta$ , and  $\varphi$  are bound to each other and drop faster in the residual balanced case (Fig. 6b) altogether for an unfavorable choice of  $\rho = \eta = \varphi = 1$ . Despite all three  $R_{rel}$ -residuals being significantly lower from the start for the ADMM-TGV, the  $S_{rel}$ -residuals do not decrease at all. As all residuals must vanish, convergence is slowed down significantly for the original algorithm. In Fig. 6 a and b, we only display the relative residuals, as they give a better impression on the development of the algorithm. For completeness, in Fig. 6c, again, both original and relative distances from convergence are shown. Here, the distance  $D^{k+1} = \|\mathbf{R}_\rho^{k+1}\|_F^2 + \|\mathbf{S}_\rho^{k+1}\|_F^2 + \|\mathbf{R}_\eta^{k+1}\|_F^2 + \|\mathbf{S}_\eta^{k+1}\|_F^2 + \|\mathbf{R}_\varphi^{k+1}\|_F^2 + \|\mathbf{S}_\varphi^{k+1}\|_F^2$  is extended by the residuals of  $\varphi$ . It can be seen that the distance from convergence vanishes faster with residual balancing overall. The same behavior is shown in Fig. 6d for the NormMSE. Interestingly, both the distance from convergence  $D$  and the NormMSE increase significantly within the first five iterations of the ADMM-RBTGV, as the  $R_{rel}$ -residuals rise. As the log-likelihood functional for Poisson noise (3), which is tied to the original image, is influenced by the residual balancing, the difference between the new iteration of  $\mathbf{x}^k$  and the original image is significantly increased. But the increase is only transient as with increasing iterations, the average  $\tau$  converges to 1 due to increased intervals as with balancing steps. After approximately 16 iterations, convergence is close for the ADMM-RBTGV (indicated as the green dotted line within the green optimum zone in Fig. 6d), while the original ADMM-TGV has not



**Fig. 4** **a** Gray scaled test image “onion” from Matlab. The pixel intensities were multiplied by a factor of 10. **b** Image with pure Poisson noise ( $\sigma_{mean} \approx 31.7$ ) added after convolution with a 2D Gaussian kernel (FWHM = 1 Pixel). **c** The result of the ADMM-TGV and **d** the ADMM-RBTGV algorithm after 1000 iterations, with unfavorable penalty parameters  $\rho = \eta = \varphi = 1$  and optimal regularization parameters  $\lambda_0 = \lambda_1 \approx 0.4/\omega\sigma$  as starting values, respectively. According to (68), the attenuation  $\omega$  amounts to  $\omega \approx 2.5$



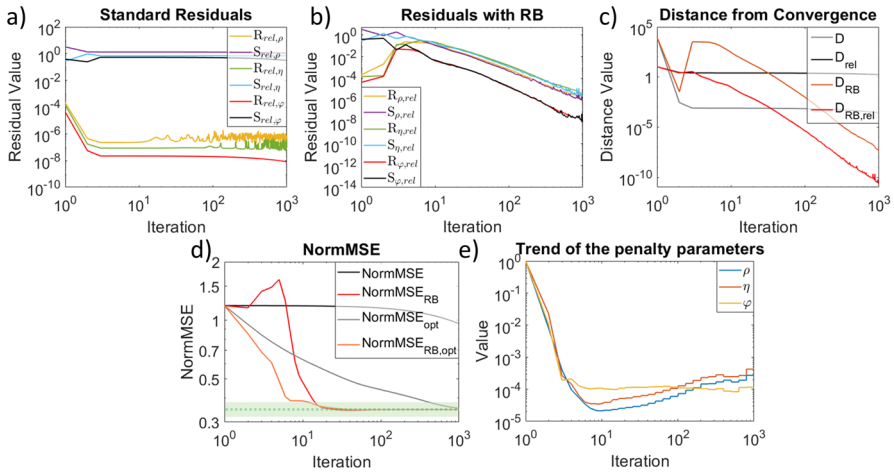
**Fig. 5** NormMSE maps, with the individual pixels showing the deconvolution results under Poisson noise with a Gaussian kernel (FWHM = 1 Pixel) of the respective  $(\rho = \eta = \varphi | \lambda_0 = \lambda_1)$ - pairs. In the upper row, the results of the original ADMM-TGV are shown after **a** 10, **b** 100, and **c** 1000 iterations. In the lower row **d–f**, the results of the residual balanced ADMM-RBTGV are displayed at the same iteration stages. The two red lines in every map indicate the region, in which the algorithm eventually improves the image quality compared to the initial noisy image, and the dashed line indicates the  $\lambda_0 = \lambda_1$ - value in units of  $1/\omega\sigma$ , with the standard deviation  $\sigma = 31.7$  of the noise and the attenuation  $\omega \approx 2.5$ , at which results are optimal ( $\lambda_{opt} \approx 0.4/\omega\sigma$ )

made significant progress. Even for the choice of a close to optimal (denoted as opt)  $\rho = \eta = \varphi = 6.3 \cdot 10^{-3}$ , the original ADMM-TGV (gray) is slower than the residual balanced ADMM-RBTGV (orange), since all three penalty parameters are identically chosen and thus suboptimal. Finding the optimal penalty parameters often is time-consuming, whereas residual balancing is fully automated and allows to quickly adapt all penalty parameters to optimal values. Figure 6e shows the development of  $\rho$ ,  $\eta$ , and  $\varphi$  due to the residual balancing, indicating that the initial values were chosen too high.

### 4.3 Optimality for $\lambda_0$ and $\lambda_1$

Having established the residual balancing for Gaussian and Poisson problems,  $\rho$ ,  $\eta$ , and  $\varphi$  are no longer necessary to be chosen by the user, but instead are automatically set by the algorithm. The parameter choice thus reduces to  $\lambda_0$  and  $\lambda_1$  to achieve optimality. Therefore, the NormMSE maps for Gaussian denoising and Poisson deconvolution are displayed in Fig. 7 a and c for pairs of  $\lambda_1$  and  $\lambda_0$ , with the starting values fixed to  $\rho = 1$ ,  $\eta = 1$  and  $\varphi = 1$  for the Poisson case.

Both maps exhibit two branches of optimality forming an L-shape. The first is connected to the value for  $\lambda_0$ , which controls the impact of the derivatives **D** acting on the flat parts of the image, and the second is connected to the  $\lambda_1$  value controlling the derivatives **G** acting like a second-order derivative applied to the sloped regions of the image. Depending on the features of the image, both parameters have to be chosen



**Fig. 6** **a** Trend of the relative  $R_{rel}$  and  $S_{rel}$  residuals (described in Sect. 3.1 and Sect. 3.2) connected to  $\rho$ ,  $\eta$ , and  $\varphi$  belonging to the standard ADMM-TGV algorithm for deconvolution under Poisson noise (Fig. 5d). Unfavorable starting values of  $\rho = \eta = \varphi = 1$  and optimal values for  $\lambda_0 = \lambda_1 = \lambda_{opt} \approx 0.4/\omega\sigma$  were chosen to compare with **b** the residuals of the residual balanced ADMM-RBTGV denoted as RB. The optimal values for  $\lambda_0 = \lambda_1$  were found as the column with minimal NormMSE values in Fig. 5d. It can be seen that for the residual balancing, the respective relative  $R_{rel}$  and  $S_{rel}$  residuals for  $\rho$ ,  $\eta$ , and  $\varphi$  stay bound together and drop faster, whereas the relative residuals of the original algorithm stagnate. Initially, this leads to a rise in all three  $R_{rel}$ - residuals. **c** Combining all six residuals to a distance measure  $D$  confirms the impression of a faster drop rate for the residual balancing. However,  $D_{RB}$  and  $D_{RB,rel}$  show a transient increase in iteration three, where the increase in  $D_{RB}$  is more pronounced. **d** The NormMSE shows that for the residual balancing, the respective relative  $R_{rel}$  and  $S_{rel}$  residuals for  $\rho$ ,  $\eta$ , and  $\varphi$  stay bound together and drop faster, whereas the relative residuals of the original algorithm stagnate. Initially, this leads to a rise in all three  $R_{rel}$ - residuals. **c** Combining all six residuals to a distance measure  $D$  confirms the impression of a faster drop rate for the residual balancing. However,  $D_{RB}$  and  $D_{RB,rel}$  show a transient increase in iteration three, where the increase in  $D_{RB}$  is more pronounced. **d** The NormMSE shows that for the residual balancing (orange), as it approaches the optimal zone (green) of the converged result (green dotted line) from iteration 16 on. The NormMSE values for the “optimal” choice of  $\rho = \eta = \varphi = 1 \cdot 10^{(-3)}$  converges a lot slower in the original algorithm (gray) than for residual balancing (orange). Like the initial rise of the distance measure  $D$  suggests, the NormMSE initially increases before seeing a fast convergence. **e** The development of  $\rho$ ,  $\eta$ , and  $\varphi$  in the course of the iteration

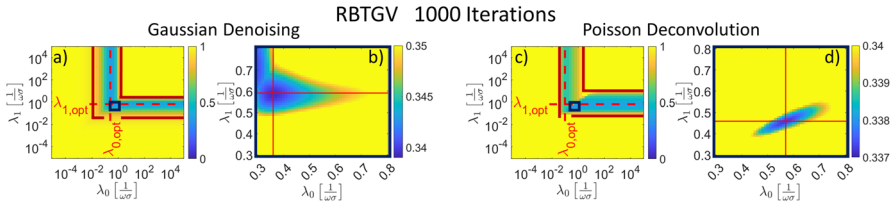
carefully and therefore typically require experience.

Since the chosen image comprises of flat and sloped features, optimality is influenced by the choice of both, the  $\lambda_0$  and  $\lambda_1$ -parameter. As long as  $\lambda_0$  is above a certain threshold of  $2/\omega\sigma$ , the result dominantly depends on the  $\lambda_1$ - parameter. The same is true as long as  $\lambda_1$  is above  $2/\omega\sigma$ , vice versa. Raising either  $\lambda_0$  or  $\lambda_1$  to unreasonably high numbers leads to large distances to the image. To reduce the distance, the filter masks adapt and minimize the parts of the image it is applied to. Thus, only the other derivative takes part in the minimization.

The optima of both branches differ significantly from 1 in Fig. 7 a and c, which is an indication that neither first- nor second-order TV alone is optimal. But  $\lambda_1$  being slightly closer to 1 than  $\lambda_0$  in a and by much in c shows that it is better suited and comes with lower cost. While in a, both branches are equal, the difference between both in c is large which shows as the much darker color of the  $\lambda_1$  branch.

As the biggest advantage of TGV is the mixture between both orders of total variation, it appears that when both types of structure, piece-wise flat and sloped, are combined in one image, the optimal  $\lambda_0, \lambda_1$  pair is found in the overlap region of both branches.





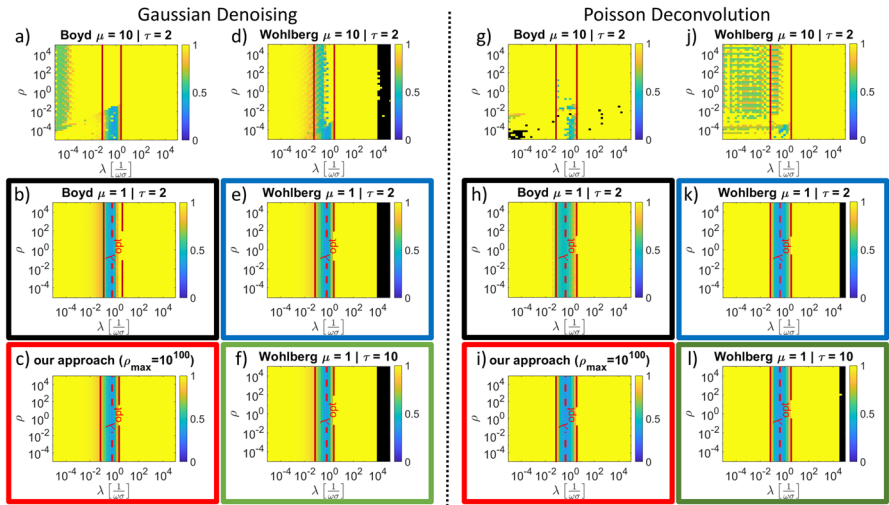
**Fig. 7** NormMSE as a function of  $\lambda_0$  and  $\lambda_1$  after applying 1000 iterations of RBTGV for Gaussian denoising **a** of Fig. 1b, with the zoomed extract marked with a blue box in **b**. The same for Poisson deconvolution **c** of Fig. 4b, with the zoomed extract marked with a blue box in **d**. The red lines in **a** and **c** indicate the region, in which the algorithm eventually improves the image quality compared to the initial noisy image. For both, **a** Gaussian denoising with the noise standard deviation  $\sigma = 50$  and the attenuation of  $\omega = 1$  as well as **c** Poisson deconvolution with  $\sigma \approx 31.7$  and the attenuation of  $\omega \approx 2.5$ , two branches occur forming an L-shape at which results are better than before. The first branch is connected to the  $\lambda_0$  balancing parameter for the first order total variation, and the second branch is connected to  $\lambda_1$ ; the dotted red line indicates the position of the optimal  $\lambda_{0,opt}$  and  $\lambda_{1,opt}$  value connected to either branch. For **a** Gaussian denoising, these values are  $(\lambda_{0,opt} | \lambda_{1,opt}) \approx (0.25 | 0.63) 1/\omega\sigma$  and for **c** Poisson deconvolution  $(\lambda_{0,opt} | \lambda_{1,opt}) \approx (0.1 | 0.63) 1/\omega\sigma$ . **b** and **d** show the overlap region of the branches for Gaussian denoising and Poisson deconvolution, respectively. Here, the red cross indicates the optimal point, at which the overall minimum is found. For **b** Gaussian denoising, the optimal value is found at  $(\lambda_0 | \lambda_1)_{opt} \approx (0.36 | 0.59) 1/\omega\sigma$ . For **d** Poisson deconvolution, the optimal value is found at  $(\lambda_0 | \lambda_1)_{opt} \approx (0.57 | 0.46) 1/\omega\sigma$

Since this overlap is only possible with the TGV approach, it shows the advantage over using only simple first- or second-order TV approaches. Figures 7 b and d display the respective overlap regions, where the optimum values can be found, as indicated by the red cross.

It appears that for both denoising and deconvolution, the optimum values for  $\lambda_0$  and  $\lambda_1$  add up to one, approximately. This indicates both are dependent on  $1/\omega\sigma$  introduced in (67) and (68). The clear difference between both sets for Gaussian denoising  $(\lambda_0 | \lambda_1)_{opt} \approx (0.36 | 0.59) 1/\omega\sigma$  in b and for Poisson deconvolution  $(\lambda_0 | \lambda_1)_{opt} \approx (0.57 | 0.46) 1/\omega\sigma$  in d is an indication that the balancing between  $\|\mathbf{z}_1\|_F$  and  $\|\mathbf{z}_2\|_F$  does not solely depend on the original image, as it is identical for both restorations, but rather on the significant features, which can be distinguished from the noise. This produces a dilemma in scientific applications that cannot be resolved, as features are not only unknown, but also the main subject of the debate, to which the restoration should add to. For an unbiased choice, we propose to weight both  $\lambda_0$  and  $\lambda_1$  equally with  $1/2\omega\sigma$ , since this choice will always increase the outcome quality significantly, while preserving an unbiased and thus scientific approach to the data.

#### 4.4 Comparison of residual balancing strategies

To further show the benefit of using our approach, it is tested against the former approaches of Boyd [22] and Wohlberg [36] (see (24) and (27)) in Fig. 8 for both denoising with Gaussian noise (a–f) and deconvolution with Poisson (g–l) noise under different parameters  $\mu$  and  $\tau$ . Boyd suggested values of  $\mu = 10$  and  $\tau = 2$  for his algorithms (see (24)) as used in Fig. 8a for Gaussian and g for Poisson noise. Further, we tested the same values for Wohlberg d and j again for both types of noise. Both approaches show unstable and inconsistent results. As residual balancing is only



**Fig. 8** NormMSE maps with the individual pixels showing the denoising results under Gaussian noise **a–f** with standard deviation  $\sigma = 50$  (same problem as Fig. 2). The approaches of Boyd [22] **a** and **b** and Wohlberg [36] **d–f** are tested with different  $\mu$  and  $\tau$  parameters (see Sect. 1.4) against **c** ours after 1000 iterations, respectively. It can be seen that only  $\mu = 1$  produces the same result for all given  $\rho = \eta$  for every  $\lambda_0 = \lambda_1$ . The black regions in the Wohlberg approach show the algorithm diverging to infinity and thus failing for the respective pairs. The colored boxes mark all results that are further analyzed in Fig. 9a, since they at least converge for optimal choices of  $\lambda$ . Images **g–l** show the deconvolution results under Poisson noise with standard deviation  $\sigma \approx 31.7$  and a 2D Gaussian Kernel with FWHM = 1 pixel (same problem as Fig. 5) under the same conditions as for the Gaussian noise aforementioned

applied when  $R$  and  $S$  residuals differ by one order of magnitude, it leads to the respective algorithms balancing the penalty parameters at different iterations depending on their initial value, which leads to different results at a given iteration. However, for this problem, we found  $\mu = 1$  as proposed by Wohlberg [36] in Fig. 8 b and h as well as e and k more convincing. Since the residual balancing is now applied every iteration, differences between the initial penalty parameters are reduced more effectively. In a test for several  $\mu$ , stability increased both for Boyds and Wohlbergs algorithms as  $\mu \rightarrow 1$ .

Increasing  $\tau$  leads to a faster adaption of all penalty parameters as the residuals can be balanced in larger steps. However, as Boyd’s algorithm does not further adapt the step size in the course of iterations, the optimal  $\rho$  and  $\eta$  can only be reached within a factor of  $\tau$ , which is why we stayed with the proposed value of  $\tau = 2$  by Boyd. Wohlberg’s algorithm, in contrast, allows to approach the optimal  $\rho$ ,  $\eta$  and  $\varphi$  since  $\tau$  is variable between 0 and its maximum value  $\tau_{max}$ .

However, choosing  $\tau$  too high leads to divergences for Wohlberg’s algorithm, marked as black areas in **d–f** and **j–l**. For both Gaussian denoising and Poisson deconvolution, this happens if the  $\lambda$ - values are estimated too high leading to over-smoothing.

Compared to Wohlberg’s algorithm, ours showed stability in both cases (Fig. 8 c and i). The main difference between Wohlbergs and our proposed algorithm here is the upper limit on  $\rho$ ,  $\eta$ , and  $\varphi$  instead of the limit on  $\tau_{max}$ , that prevents our algorithm

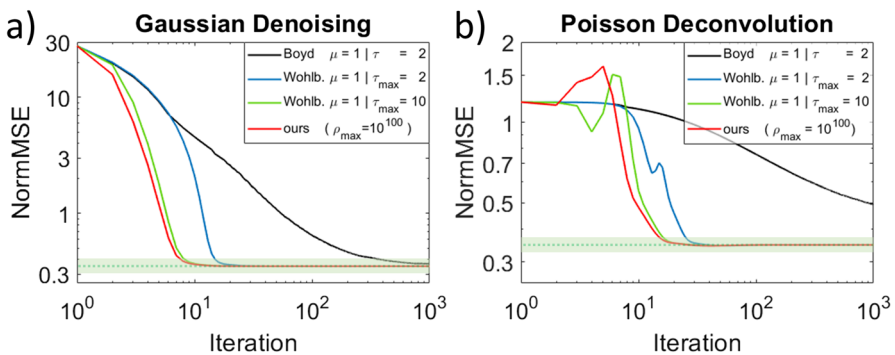
from divergence, and the increasing number of iterations between two consecutive balancing steps, which enforces convergence instead.

As Fig. 8 does not show the speed or progress of the respective algorithms, all converging strategies marked with colored boxes are tested against each other in Fig. 9 at an unfavorable  $\rho = \eta = \varphi = 1$  and  $\lambda_{opt}$  pair (similar to Fig. 3d and Fig. 6d). The graphs indicate that the approach of Boyd for Gaussian denoising (Fig. 9a) and Poisson deconvolution (Fig. 9b) is the slowest, since the optimal penalty parameters can never be reached due to the limitation of  $\tau$ , which is overcome by Wohlberg’s approach with  $\tau_{max} = 2$ . For the first six iterations, both Boyd and Wohlberg’s approach show the same NormMSE values, indicating that afterwards, the variable  $\tau$  is beneficial. Further increasing  $\tau_{max}$  leads to a faster convergence, as the maximum balancing step is increased. Our approach is superior to the others in terms of convergence speed, since we do not limit the step size at all, but only set an upper limit to  $\rho, \eta,$  and  $\varphi$ .

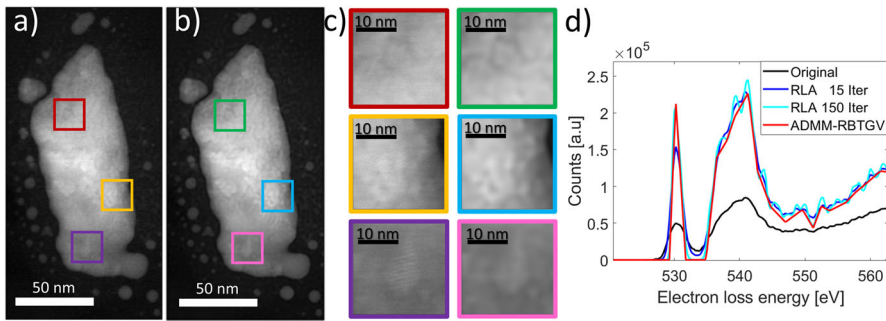
### 5 Application examples

In Fig. 10, we give two examples of applications in physics. Figure 10a shows gold nanoparticles on a silicon oxide substrate as acquired with a scanning transmission electron microscope (STEM) in annular dark field mode (ADF) with inherent blur by the point-spread-function of the electron beam and some minor defocus. Deconvolving with a more specialized and adaptable mixed Poisson-Gaussian noise model (MPG) [43] with our ADMM-RBTGV (b) improves the overall quality of the reconstruction compared to the original as can especially be seen in the zoomed extract of c, where the crystalline structures appear much clearer.

The second example in Fig. 10d demonstrates the application of the ADMM-RBTGV algorithm to 1D data (spectra), here an electron-energy loss spectrum (EELS) recorded in a STEM. EEL spectra suffer from both the finite energy width of the impinging



**Fig. 9** The graphs display the progress of the NormMSE within 1000 iterations. To show differences in the speed of convergence from the respective stable ADMM-RBTGV algorithms for **a** Gaussian and **b** Poisson noise (see Fig. 8) marked with the same line color as the boxes before. It can be seen that Boyd’s (black) approach [22] is the slowest of all for both problems. It is observed that an increase of the  $\tau_{max}$  translates into a faster convergence for Wohlberg’s approach [36] (blue and green). The fastest convergence is achieved with our algorithm (red) by only limiting  $\rho$  and  $\eta$  instead of  $\tau$



**Fig. 10** **a** Noisy image of a metallic nanoparticle acquired in a transmission electron microscope. **b** The deconvolved image with the ADMM-RBTGV after 1500 iterations. **c** Comparison of zoomed extracts of both, original and deconvolved images at the positions marked with the respective colors. **d** Electron energy loss signal from the oxygen *K*-edge of a Hematite ( $\text{Fe}_2\text{O}_3$ ) nanoparticle with only the background subtracted (black), deconvolved with the zero-loss-peak by the commonly used RLA after 15 iterations (blue), 150 iterations (cyan) iterations (red), and the ADMM-RBTGV after 1500 iterations, with a mixed Poisson-Gauss noise model [43]. It can be seen that our approach produces a much smoother outcome, due to the better-adjusted noise model. Additionally, our result leaves much less room for interpretations of additional peaks, that occur around 540 eV in both RLA results

electron beam, leading to a convolution of characteristic spectral features, and noise, owing to the limited electron dose used in a measurement. The black spectrum was recorded from a catalytic metal oxide nanoparticle near the oxygen *K*-edge, where a background was subtracted using a common power-law mode. The blue and cyan curves show the spectrum after applying the commonly used Richardson-Lucy algorithm (RLA) 15 and 150 times, respectively. Deconvolving the image with the RLA sharpens the edges, but due to a misfit in the noise model by completely ignoring Gaussian read-out noise and not regarding the change in statistics by the subtraction of the background, the RLA diverges with increasing iterations. This is why the RLA is often limited to 15 iterations in EELS applications [44]. Oscillations occurring after 150 iterations of RLA near the spectral maximum at 542 eV might be easily misinterpreted as characteristic features. At energy losses above 550 eV, RLA-treated spectra show additional oscillations with the same amplitude as the peaks around 542 eV are produced by the RLA. These artifacts are commonly known as ringing artifacts [45–47]. In comparison, the spectrum obtained by applying the ADMM-RBTGV (1500 times) nicely corresponds to the spectra expected for metal oxide particles [48–51]. Thus, by utilizing MPG with our ADMM-RBTGV, the overall quality of the reconstruction improves significantly compared to the other as the signal appears much smoother and less noisy. All in all, the examples show that ADMM-RBTGV can reliably improve image and signal quality, unbiased by the choice of input parameters by the user.

## 6 Conclusions

In this paper, a novel residual balancing approach for ADMM image denoising and deconvolution with total generalized variation is described. It is applied to standard images as well as scientific 2D and 1D data superimposed with both Gaussian- and

Poisson-type noises. The usefulness, objectivity, and fast convergence of the algorithm are demonstrated. By automatically choosing penalty parameters  $\rho$ ,  $\eta$ , and  $\varphi$  for the Poisson case and equalizing their initial choice after a few iterations, residual balancing makes the algorithm user-friendly and unbiased. Moreover, it increases the convergence speed for almost all initial choices of the penalty parameters significantly. If the optimum is already known, residual balancing is indeed slower than the original algorithm, but does not delay convergence by much. However, as the optimum is hard to guess, residual balancing provides an easy and successful way of choosing  $\rho$  and  $\eta$  in the case of the Gaussian noise ADMM and an additional  $\varphi$  for the Poisson case. It is shown that the original algorithm and the residual balanced one converge to the same result for different initial choices of the penalty parameters over several orders of magnitude for the same values for the regularization parameters  $\lambda_0$  and  $\lambda_1$ . Further, we could show that our approach of residual balancing is superior to others, as it provides the fastest convergence and is stable at all tested pairs.

As the applicability is clearly shown, this work allows to adapt the principles of residual balancing to other penalizers for image restoration in general and for different dimensionality of the data.

By reducing the set of four parameters (or five in the Poisson case) to be chosen by the user to two, optimal regions can be found by only testing  $(\lambda_0 | \lambda_1)$ -pairs. This makes visualization easier and reduces computational time for finding the optimal set.

It is shown that the regularization parameters  $\lambda_0$  connected to the first derivative and  $\lambda_1$  connected to the second derivative depend on the inverse standard deviation of the noise  $1/\omega\sigma$ , that gets attenuated by a factor of  $\omega$  in the case of a deconvolution. This factor depends on the deconvolution kernel, as it drags more noise into a single pixel. Selecting either  $\lambda_0$  or  $\lambda_1$  too high while keeping the other close to the optimum leads to the filter masks adapting and minimizing the influence of the respective first or second matrix of derivatives **D** or **G**. The conclusion drawn from this is that TGV can either converge to results of the standard first-order TV or the second-order TV. However, as most images consist of both flat and sloped parts, optimality is found in the overlap of both  $\lambda_0$  and  $\lambda_1$  regions. Here, we could show that both regularization parameters  $\lambda_0$  and  $\lambda_1$  must add up to a total value of  $1/\omega\sigma$  to achieve optimality. The ratio between both depends on the significant features in the image, which can be distinguished from the noise. Thus, for the same image under different noise conditions, the optimality of these parameters is not identical, but changes significantly.

Finally, with our work, the algorithm can be set up such that it operates on objective criteria and not on the arbitrary choice of user inputs: The penalty parameters are chosen automatically on a mathematical foundation, and the regularization parameters  $\lambda_0$  and  $\lambda_1$  can be chosen equal to  $1/2\omega\sigma$ , such that they only depend on the noise standard deviation  $\sigma$ . Since the features in an image are not known in a scientific context, the true optimality of the ratio of both parameters cannot be achieved without introducing a certain bias. But fixing the ratio of both regularization parameters still improves images significantly, since both parameters continue to be located within the optimality region. This allows for an unbiased approach on the restoration of experimental data as there is no room for altering the results. With sufficient knowledge of the noise in an experiment, which can be found by carefully investigating the detector, our work enables the use of the ADMM-based image restoration algorithms for scientific tasks.

**Acknowledgements** The authors thank the associated editor and anonymous reviewers for the helpful comments and suggestions.

**Author contribution** All authors contributed to the study conception and design. The manuscript was written by CZ. All authors read and approved the final manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Availability of data and materials** Not applicable

**Code availability** Not applicable

## Declarations

**Ethics approval** Not applicable

**Consent for publication** All authors give their consent for publication.

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Sur, F., Grédiac, M.: Measuring the noise of digital imaging sensors by stacking raw images affected by vibrations and illumination flickering. *SIAM J. Imaging Sci.* **3**(1), 611–643 (2015)
2. Wen, Y.W., Ng, M.K., Ching, W.K.: Iterative algorithms based on decoupling of deblurring and denoising for image restoration. *SIAM J. Sci. Comput.* **30**(5), 2655–2674 (2008)
3. Cohen, R., Elad, M., Milanfar, P.: Regularization by denoising via fixed-point projection (RED-PRO). *SIAM J. Imaging Sci.* **14**(3), 1374–1406 (2021)
4. Vogel, C.R., Oman, M.E.: Iterative methods for total variation denoising. *SIAM J. Sci. Comput.* **17**(1), 227–238 (1996)
5. Chan, S.H., Wang, X., Elgandy, O.A.: Plug-and-play ADMM for image restoration: fixed-point convergence and applications. *IEEE Trans. Comput. Imaging* **3**(1), 84–98 (2017)
6. Bredies, K., Sun, H.: Preconditioned Douglas-Rachford splitting methods for convex-concave saddlepoint problems. *SIAM J. Numer. Anal.* **53**(1), 421–444 (2015)
7. Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *ESAIM Math. Model. Numer. Anal.* **9**(R2), 41–76 (1975)
8. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **2**(1), 17–40 (1976)
9. Goldstein, T., O'Donoghue, B., Setzer, S., Baraniuk, R.: Fast alternating direction optimization methods. *SIAM J. Imaging Sci.* **7**(3), 1588–1623 (2014)
10. Figueiredo, M.: On the use of ADMM for imaging inverse problems: the pros and cons of matrix inversions. In: *Topics in Applied Analysis and Optimisation*, pp. 159–181. Springer International Publishing, ISBN: 978-3-030-33115-3 (2019)

11. Cheon, W., Min, B.J., Seo, Y.S., Lee, H.: Non-blind deconvolution with an alternating direction method of multipliers (ADMM) after noise reduction in nondestructive testing. *JINST* **14**, P11032 (2019)
12. Wang, X., Chan, S.H.: Parameter-free plug-and-play ADMM for image restoration. *IEEE Proc. IEEE Int Conf Acoust Speech Signal Process (ICASSP)*, pp. 1323–1327 (2017)
13. Xie, W.S., Yang, Y.F., Zhou, B.: An ADMM algorithm for second-order TV-based MR image reconstruction. *Numer Algor.* **67**, 827–843 (2014)
14. Fang, Z., Liming, T., Liang, W., Liu, H.: A nonconvex  $TV_q$ - $\ell_1$  regularization model and the ADMM based algorithm. *Sci. Rep.* **12**, 7942 (2022)
15. Eksioğlu, E.M., Tanc, A.K.: Denoising AMP for MRI reconstruction: BM3D-AMP-MRI. *SIAM J. Imaging Sci.* **11**(3), 2090–2109 (2018)
16. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1–4), 259–268 (1992)
17. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
18. Knoll, F., Bredies, K., Pock, T., Stollberger, R.: Second order total generalized variation (TGV) for MRI. *Magn. Reson. Med.* **65**(2), 480–491 (2011)
19. Ghadimi, E., Teixeira, A., Shames, I., Johansson, M.: Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. *IEEE Trans. Automat. Contr.* **60**(3), 644–658 (2015)
20. Chorfi, A., Koko, J.: Alternating direction method of multiplier for the unilateral contact problem with an automatic penalty parameter selection. *Appl. Math. Model.* **78**, 706–723 (2020)
21. Lin, Y., Wohlberg, B., Vesselinov, V.: ADMM penalty parameter selection with Krylov subspace recycling technique for sparse coding. In: *IEEE International Conference on Image Processing (ICIP)*, pp. 1945–1949 (2017)
22. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1) (2010)
23. Sanders, T.: Parameter selection for HOTV regularization. *Appl. Numer. Math.* **125**, 1–9 (2018)
24. Wen, Y.W., Chan, R.H.: Parameter selection for total-variation-based image restoration using discrepancy principle. *IEEE Trans. Image Process* **21**(4), 1770–1781 (2012)
25. Das, P.P., Guzzinati, G., Coll, C., Gomez Perez, A., Nicolopoulos, S., Estrade, S., Peiro, F., Verbeeck, J., Zompra, A.A., Galanis, A.S.: Reliable characterization of organic & pharmaceutical compounds with high resolution monochromated eel spectroscopy. *Polymers* **12**, 1434 (2020)
26. Brokkelkamp, A., ter Hoeve, J., Postmes, I., van Heijst, S.E., Maduro, L., Davydov, A.V., Krylyuk, S., Rojo, J., Conesa-Boj, S.: Spatially resolved band gap and dielectric function in two-dimensional materials from electron energy loss spectroscopy. *J. Phys. Chem. A* **126**(7), 1255–1262 (2022)
27. Eljarrat, A., López-Conesa, L., Estradé, S., Peiró, F.: Electron energy loss spectroscopy on semiconductor heterostructures for optoelectronics and photonics applications. *J. Microsc* **262**, 142–150 (2016)
28. Lajaunie, L., Ramasubramaniam, A., Panchakarla, L.S., Arenal, R.: Optoelectronic properties of calcium cobalt oxide misfit nanotubes. *Appl. Phys. Lett.* **16**;113(3), 031102 (2018)
29. Song, Y., Li, H., Zhai, G., He, Y., Bian, S., Zhou, W.: Comparison of multichannel signal deconvolution algorithms in airborne LiDAR bathymetry based on wavelet transform. *Sci. Rep.* **11**, 16988 (2021)
30. Perez, V., Chang, B.J., Stelzer, E.: Optimal 2D-SIM reconstruction by two filtering steps with Richardson-Lucy deconvolution. *Sci. Rep.* **6**, 37149 (2016)
31. Ishizuka, A., et al.: Improving the depth resolution of STEM-ADF sectioning by 3D deconvolution. *Microscopy* **70**(2), 241–249 (2021)
32. Bayes, T.: LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. In: *Philosophical Transactions of the Royal Society of London*, Vol. 53, pp. 370–418 (1763)
33. Held, L., Sabanés Bové, D.: *Applied statistical inference: likelihood and Bayes*. Springer, Berlin Heidelberg (2014)
34. Ikoma, H., Broxton, M., Kudo, T., Wetzstein, G.: A convex 3D deconvolution algorithm for low photon count fluorescence imaging supplementary material. *Sci. Rep.* **8**, 11489 (2018)
35. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. *Proc. Second Berkeley Symp. Math. Statist. Prob.* (Univ. of Calif. Press), 481–492 (1951)
36. Wohlberg, B.: ADMM penalty parameter selection by residual balancing. [arXiv:1704.06209v1](https://arxiv.org/abs/1704.06209v1)
37. He, B.S., Yang, H., Wang, S.L.: Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *J. Optim. Theory Appl.* **106**(2), 337–356 (2000)

38. Liu, Z., Hansson, A., Vandenberghe, L.: Nuclear norm system identification with missing inputs and outputs. *Syst. Control. Lett.* **62**(8), 605–612 (2013)
39. Iordache, M.D., Bioucas-Dias, J.M., Plaza, A.: Collaborative sparse regression for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **52**(1), 341–354 (2014)
40. Wohlberg, B.: Efficient convolutional sparse coding. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7173–7177 (2014)
41. Weller, D.S., Pnueli, A., Radzyner, O., Divon, G., Eldar, Y.C., Fessler, J.A.: Phase retrieval of sparse signals using optimization transfer and ADMM. In: *IEEE International Conference on Image Processing (ICIP)*, pp. 1342–1346 (2014)
42. Shirai, K., Okuda, M.: FFT based solution for multivariable  $L_2$  equations using KKT system via FFT and efficient pixel-wise inverse calculation. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2629–2633 (2014)
43. Ding, Q., Long, Y., Zhang, X., Fessler, J.A.: Statistical image reconstruction using mixed Poisson-Gaussian noise model for X-ray CT. [arXiv:1801.09533](https://arxiv.org/abs/1801.09533)
44. Egerton, R.F., Wang, F., Malac, M., Moreno, M.S., Hofer, F.: Fourier-ratio deconvolution and its Bayesian equivalent. *Micron* **39**(6), 642–647 (2008)
45. Cui, G., Feng, H., Xu, Z., Li, Q., Chen, Y.: A modified Richardson-Lucy algorithm for single image with adaptive reference maps. *Opt. Laser Technol.* **58**, 100–109 (2014)
46. Zeng, G.L.: Gibbs artifact reduction by nonnegativity constraint. *J. Nucl. Med. Technol.* **39**(3), 213–9 (2011)
47. Wang, W., Wang, W., Yin, J.: A bilateral filtering based ringing elimination approach for motion-blurred restoration image. *Curr. Opt. Photonics* **4**(3), 200–209 (2020)
48. Wang, C., Yiqian Wang, Y., Liu, X., Yang, H., Sun, J., Yuan, L., Zhou, G., Rosei, F.: Structure versus properties in  $\alpha$ -Fe<sub>2</sub>O<sub>3</sub> nanowires and nanoblades. *Nanotechnology* **27**(3), 035702 (2015)
49. Cortés-Llanos, B., Serrano, A., Muñoz-Noval, A., Urones-Garrote, E., del Campo, A., Marco, J.F., Ayuso-Sacido, A., Pérez, L.: Thermal route for the synthesis of maghemite/hematite core/shell nanowires. *J. Phys. Chem. C* **121**(41), 23158–23165
50. Guo, Q., Luo, H., Zhang, J., Ruan, Q., Prakash Periasamy, A., Fang, Y., Xie, Z., Li, X., Wang, X., Tang, J., Briscoe, J., Titirici, M., Jorge, A.B.: The role of carbon dots - derived underlayer in hematite photoanodes. *Nanoscale* **12**(39), 20220–20229 (2020)
51. Ullrich, A., Rahman, M.M., Longo, P., Horn, S.: Synthesis and high-resolution structural and chemical analysis of iron-manganese-oxide core-shell nanocubes. *Sci. Rep.* **9**, 19264 (2019)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.