

# Performance of First-Order Algorithms for TV Penalized Weighted Least-Squares Denoising Problem

Alex Sawatzky

Institute for Computational and Applied Mathematics,  
University of Münster, Einsteinstr. 62, 48149 Münster, Germany  
<http://imaging.uni-muenster.de/>

**Abstract.** Denoising of images perturbed by non-standard noise models (e.g., Poisson or Gamma noise) can be often realized by a sequence of penalized weighted least-squares minimization problems. In the recent past, a variety of first-order algorithms have been proposed for convex problems but their efficiency is usually tested with the classical least-squares data fidelity term. Thus, in this manuscript, first-order state-of-the-art computational schemes are applied on a total variation penalized weighted least-squares denoising problem and their performance is evaluated on numerical examples simulating a Poisson noise perturbation.

**Keywords:** Image denoising, weighted least-squares, total variation, first-order algorithm, primal-dual algorithm, alternating direction method of multipliers, Chambolle-Pock.

## 1 Introduction

This work considers the discrete variational denoising problem

$$\min_{u \in C} \frac{1}{2} \|u - f\|_{\frac{1}{w}}^2 + \alpha \|\nabla u\|_1, \quad \alpha > 0, \quad (1)$$

where  $f$  and  $u$  are  $d$ -dimensional images ( $d \in \{2, 3\}$  typically) and denote the observed noisy image and the desired solution, respectively. The data fidelity term  $\|x\|_z^2 = \langle zx, x \rangle$  with  $z$  being positive denotes a weighted scalar product and the penalty term  $\|\nabla u\|_1$  is a discrete version of the total variation (TV) seminorm. The set  $C$  is a non-empty closed convex set which may include additional constraints such as non-negativity.

The restoration of noisy images using TV penalty has grown in popularity due to its ability to reduce noise efficiently and to preserve almost sharp edges in the image. One of the most famous applications of TV regularization in image processing is the ROF model proposed by Rudin, Osher, and Fatemi [1] for image denoising and is defined as (1) with  $w \equiv 1$ . To solve (1), we restrict the attention to first-order methods which use the (sub)gradient information of the cost function only. Algorithms of this type have become popular in the recent past such

as proximal algorithms [2], projected gradient algorithms [3,4], or primal-dual methods [5,6,7]. In this work, the state-of-the-art numerical schemes proposed for the ROF model will be modified to address (1) and their performance will be compared with respect to the numerical efficiency.

The problem (1) appears in different image processing tasks and a few of which are shortly discussed in the following.

*Additive Multivariate Gaussian Noise Model:* From the view of statistical (Bayesian) modeling, the ROF problem results from the assumption of additive Gaussian noise model  $f = \bar{u} + \eta$ , where  $\bar{u}$  is the exact image and  $\eta$  is a Gaussian-distributed random variable with a scalar variance. However, if the characteristic of noise varies across the image, the variance of  $\eta$  is multivariate and the statistical modeling approach leads to the weighted least-squares fidelity term as in (1) where the weight  $w$  is specified by the noise variances.

*Heteroscedastic Noise:* Diverse instruments and processes lead to measurements disturbed by heteroscedastic noise, i.e. the noise characteristics depend on the (unknown) signal, or pre-processing stages may transform a homoscedastic noise to a heteroscedastic one. Such noise characteristics can be found, e.g., in digital cameras [8,9] or ultrasound imaging [10,11]. With notations from above, such a noise can be modeled as  $f = \bar{u} + \eta(\bar{u})$  with a signal dependent variance of  $\eta$  and proper fidelity weights in (1) can stabilize the denoising process.

*FB-EM-TV Reconstruction Framework:* In [12], a numerical framework has been proposed for inverse problems with Poisson noise. The numerical scheme is realized by alternating a classical expectation-maximization (EM) reconstruction step and solving the denoising problem (1) with an appropriate setting of  $f$ ,  $w$ , and  $\alpha$ . If the FB-EM-TV framework is used to denoise a Poisson perturbed image, the iteration method results in a sequence of (1) [13].

*Forward-Backward Splitting Algorithm with Variable Metric Strategies:* A large class of reconstruction problems can be formulated as a problem minimizing a function of the form  $F + R$ , where  $F$  is differentiable and  $R$  is a convex and proper lower semicontinuous function. In context of statistically-principled reconstruction problems,  $F$  corresponds to the data fidelity term and  $R$  is the regularization energy. A popular approach to solve such a problem is the forward-backward (FB) splitting algorithm (also known as proximal-gradient algorithm), which alternates a gradient descent step on  $F$  and a proximal step on  $R$  (see, e.g., [14] and references therein). One way to accelerate the convergence of FB algorithm is based on a variable metric strategy leading to a preconditioned gradient descent on  $F$ . In this case, the proximal step on  $R$  minimizes a penalized weighted least-squares cost function as in (1) [14]. The FB-EM-TV framework mentioned above is, e.g., an instance of such a FB strategy (see [12]).

*Newton-Type Methods for Non-Gaussian Denoising Problems:* Bayesian modeling of denoising problems with non-Gaussian noise models leads in general to strongly non-linear data fidelity terms resulting in issues in the computation of minimizers. Such issues can be overcome by a local approximation of non-linear terms with quadratic functions leading to a sequence of reweighted least-squares fidelity terms, see, e.g., [15] in case of Poisson noise or a specific speckle noise

model used to describe the noise properties in the medical ultrasound imaging. Further non-classical examples can be found in [16].

## 2 Methods

### 2.1 Notations

In this work, we consider a  $d$ -dimensional image as a matrix on a regular grid of  $N_1 \times \dots \times N_d$  points with index set  $\mathcal{I} = \{(i_1, \dots, i_d) : 1 \leq i_k \leq N_k, 1 \leq k \leq d\}$  and denote with  $h_k$  the stepsize of the image grid in the  $k$ -th direction. The weighted scalar product in  $\mathbb{R}^{|\mathcal{I}|}$  is defined as  $\langle zx, y \rangle_{\mathbb{R}^{|\mathcal{I}|}} = \sum_{i \in \mathcal{I}} z_i x_i y_i$ . For the discrete gradient operator  $\nabla$  with  $(\nabla u)_i = ((\nabla u)_i^k)_{k=1}^d$ ,  $i \in \mathcal{I}$ , the standard forward finite differences with Neumann boundary conditions are used,

$$(\nabla u)_{i_1, \dots, i_d}^k = \begin{cases} (u_{i_1, \dots, i_k+1, \dots, i_d} - u_{i_1, \dots, i_k, \dots, i_d})/h_k, & \text{if } i_k < N_k, \\ 0, & \text{if } i_k = N_k, \end{cases}$$

and the discrete isotropic total variation is defined as  $\|\nabla u\|_1 = \sum_{i \in \mathcal{I}} |(\nabla u)_i|$  with  $|\cdot|$  being the Euclidean vector norm. The discrete divergence operator  $\text{div}$  is chosen to be adjoint to the gradient operator (i.e., it holds  $\nabla^* = -\text{div}$ ) and characterized through the identity  $\langle \nabla u, g \rangle_{\mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d} = -\langle u, \text{div} g \rangle_{\mathbb{R}^{|\mathcal{I}|}}$ , where the scalar product in  $\mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d$  is set as  $\langle p, g \rangle_{\mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d} = \sum_{i \in \mathcal{I}} \sum_{k=1}^d p_i^k g_i^k$ . The shrinkage operator *shrink* required below is defined as

$$\text{shrink}(x, \gamma) = (x/|x|) \max(|x| - \gamma, 0), \quad x \in \mathbb{R}^d, \quad \gamma > 0. \quad (2)$$

Finally,  $\mathcal{P}_C(x)$  denotes the projection of  $x$  into the convex set  $C$  and, as long as not mentioned explicitly, all elementary operations as addition, multiplication, or maximum operations on  $\mathbb{R}^{|\mathcal{I}|}$  or  $\mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d$  as well as the Euclidean vector norm  $|\cdot|$  on  $\mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d$  have to be interpreted pointwise (i.e., for each  $i \in \mathcal{I}$ ).

### 2.2 Accelerated Projected (Sub)Gradient Descent Algorithm

First, we use a modified variant of the accelerated projected (sub)gradient descent algorithm proposed in [3]. In contrast to the original Chambolle's algorithm

---

#### Algorithm 1. Accelerated Projected (Sub)Gradient Descent Algorithm

---

1. **Initialization:**  $t^0 = 1$ ,  $g^0 = r^0 = 0 \in \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d$ ,  $0 < \tau \leq \left(\alpha \|w\|_\infty 4 \sum_{k=1}^d \frac{1}{h_k^2}\right)^{-1}$

2. **Iteration:** For  $k = 0, 1, \dots$

$$g^{k+1} = \frac{r^k - \tau \nabla (\mathcal{P}_C(f - \alpha w \text{div} r^k))}{\max(1, |r^k - \tau \nabla (\mathcal{P}_C(f - \alpha w \text{div} r^k))|)}$$

$$t^{k+1} = \left(1 + \sqrt{1 + 4(t^k)^2}\right) / 2$$

$$r^{k+1} = g^{k+1} + \frac{t^k - 1}{t^{k+1}} (g^{k+1} - g^k)$$

3. **Return**  $u = \mathcal{P}_C(f - \alpha w \text{div} g^{k+1})$

---

---

**Algorithm 2.** Split Bregman Algorithm

---

1. **Initialization:**  $u^0 = f, v^0 = \nabla f, \lambda_v^0 = 0 \in \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d, \mu_v > 0$
  2. **Iteration:** For  $k = 0, 1, \dots$ 
    - (i) Solve  $u^{k+1} - \mu_v w \Delta u^{k+1} = f - w \operatorname{div}(\lambda_v^k + \mu_v v^k)$  s.t.  $u^{k+1} \in C$
    - (ii)  $v_i^{k+1} = \operatorname{shrink}((\nabla u^{k+1})_i - (\mu_v)^{-1}(\lambda_v^k)_i, \alpha/\mu_v), \quad i \in \mathcal{I}$  ▷ (2)
    - (iii)  $\lambda_v^{k+1} = \lambda_v^k + \mu_v (v^{k+1} - \nabla u^{k+1})$
  3. **Return**  $u^{k+1}$
- 

---

**Algorithm 3.** Alternating Direction Method of Multipliers

---

1. **Initialization:**  $\tilde{u}^0 = f, v^0 = \nabla f, \lambda_{\tilde{u}}^0 = 0 \in \mathbb{R}^{|\mathcal{I}|}, \lambda_v^0 = 0 \in \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^d, \mu_{\tilde{u}}, \mu_v > 0$
  2. **Iteration:** For  $k = 0, 1, \dots$ 
    - (i) Solve  $\mu_{\tilde{u}} u^{k+1} - \mu_v \Delta u^{k+1} = \lambda_{\tilde{u}}^k + \mu_{\tilde{u}} \tilde{u}^k - \operatorname{div}(\lambda_v^k + \mu_v v^k)$
    - (ii)  $\tilde{u}^{k+1} = \mathcal{P}_C(f + w(\mu_{\tilde{u}} u^{k+1} - \lambda_{\tilde{u}}^k)/(1 + \mu_{\tilde{u}} w))$
    - (iii)  $v_i^{k+1} = \operatorname{shrink}((\nabla u^{k+1})_i - (\mu_v)^{-1}(\lambda_v^k)_i, \alpha/\mu_v), \quad i \in \mathcal{I}$  ▷ (2)
    - (iv)  $\lambda_{\tilde{u}}^{k+1} = \lambda_{\tilde{u}}^k + \mu_{\tilde{u}} (\tilde{u}^{k+1} - u^{k+1}), \lambda_v^{k+1} = \lambda_v^k + \mu_v (v^{k+1} - \nabla u^{k+1})$
  3. **Return**  $\tilde{u}^{k+1}$
- 

[17] that suffers from slow linear convergence [18], the accelerated variant ensures a quadratic convergence rate which is "optimal" for the class of first-order methods with Lipschitz differentiable functionals [19,20]. The modification of [3] to (1) is summarized in *Algorithm 1*. The restriction on the damping parameter  $\tau$  is required to guarantee the convergence of the iteration sequence to an optimal solution, where  $4 \sum_{k=1}^d h_k^{-2}$  is the upper bound of the divergence operator.

### 2.3 Alternating Direction Method of Multipliers and Its Variants

**Split Bregman Algorithm.** In the recent past, the split Bregman algorithm [21] has become popular due to the efficiency to solve TV penalized variational problems and represents an instance of the alternating direction method of multipliers (ADMM) [5]. The central idea is to reformulate (1) to an equivalent constrained problem using an auxiliary variable  $v = \nabla u$  in the penalty term and to force this constraint by the Bregman distance. The resulting numerical scheme is summarized in *Algorithm 2*.

**Multiple Splitting Approach.** In contrast to the split Bregman algorithm, the flexibility of ADMM method allows to specify more than one auxiliary variable. An alternative to solve (1) using ADMM has been recently proposed in [12] where auxiliary variables  $\tilde{u} = u$  and  $v = \nabla u$  were introduced. More details can be found in [12] and the iteration sequence is summarized in *Algorithm 3*. This algorithm allows a more efficient computation of  $u^{k+1}$  (cf. with (i) in *Algorithm 2*) since the fidelity weight  $w$  is decoupled from the linear equation.

**Accelerated Variant of ADMM Algorithms.** In [22], the authors proposed an acceleration of ADMM method which achieves a quadratic convergence rate

---

**Algorithm 4.** Chambolle-Pock’s Primal-Dual Algorithm

---

1. **Initialization:**  $u^0 = \bar{u}^0 = f$ ,  $g^0 = 0 \in \mathbb{R}^{|I|} \times \mathbb{R}^d$ ,  $\tau^0, \sigma^0 > 0$  with  $\tau^0 \sigma^0 \|\operatorname{div}\|^2 \leq 1$
  2. **Iteration:** For  $k = 0, 1, \dots$ 
    - (i)  $g^{k+1} = (g^k + \sigma^k \nabla \bar{u}^k) / \max(1, |g^k + \sigma^k \nabla \bar{u}^k|)$
    - (ii)  $u^{k+1} = \mathcal{P}_C(\alpha w(u^k + \tau^k \operatorname{div} g^{k+1} + \tau^k f) / (\alpha w + \tau^k))$
    - (iii)  $\theta^k = 1 / \sqrt{1 + 2\gamma\tau^k}$ ,  $\tau^{k+1} = \theta^k \tau^k$ ,  $\sigma^{k+1} = \sigma^k / \theta^k$
    - (iv)  $\bar{u}^{k+1} = u^{k+1} + \theta^k (u^{k+1} - u^k)$
  3. **Return**  $u^{k+1}$
- 

in contrast to the linear convergence of the basic ADMM algorithm. Thus, we additionally modified the *Algorithms 2* and *3* corresponding to the proposed acceleration and restarted the acceleration process [22] if

$$\max((\sum_{i \in M} \|r_i^k\|^2)^{1/2}, \|r_u^k\|) - \max((\sum_{i \in M} \|r_i^{k+1}\|^2)^{1/2}, \|r_u^{k+1}\|) \leq 0$$

with  $M = \{v\}$  and  $M = \{\tilde{u}, v\}$  in case of *Algorithm 2* and *3*, respectively, and

$$r_{\tilde{u}}^k = \|\mu_{\tilde{u}}(\tilde{u}^k - u^k)\|, \quad r_v^k = \|\mu_v(v^k - \nabla u^k)\|, \tag{3a}$$

$$r_u^k = \|\mu_v \operatorname{div}(v^k - \hat{v}^k)\| \tag{3b}$$

$$r_u^k = \|\mu_{\tilde{u}}(\tilde{u}^k - \hat{\tilde{u}}^k) - \mu_v \operatorname{div}(v^k - \hat{v}^k)\| \tag{3c}$$

are the residuals for the primal and dual feasibility conditions [5,22] with  $\hat{\tilde{u}}^k$  and  $\hat{v}^k$  being the accelerated iterates as introduced in [22].

### 2.4 Chambolle-Pock’s Primal-Dual Algorithm

Recently, a first-order primal-dual hybrid algorithm has been proposed in [6] and the authors have shown its improved efficacy compared to the accelerated projected (sub)gradient descent algorithm and the split Bregman algorithm in case of the ROF model. Thus, as the fidelity term in (1) is uniformly convex, we modified the accelerated version of the primal-dual algorithm (see Algorithm 2 and Section 6.2.1 in [6]) to (1) and summarize the resulting iteration scheme in *Algorithm 4*. Finally, note that this primal-dual algorithm is closely related to the proximal ADMM method [7,23].

## 3 Numerical Results

To evaluate the performance of algorithms discussed, the following procedure was applied (cf. [6]). First, since (1) is strictly convex for  $w > 0$ , the minimizer is unique and we can run a well performing method for a very long time to compute a "ground truth" solution  $u_\alpha^*$  for a fixed  $\alpha$ . Thus, we have run *Algorithm 4* for  $N = 100000$  iterations for following reasons: (1) quadratic convergence rate is guaranteed if  $\tau^0 \sigma^0 \|\operatorname{div}\|^2 \leq 1$  is fulfilled [6] and thus  $u_\alpha^*$  is expected to be of high accuracy after  $N$  iterations; (2) all iteration steps can be solved exactly such that  $u_\alpha^*$  cannot be influenced by inexact computations required in *Algorithm 2*

(see below); (3) in contrast to *Algorithm 3*, the parameter setting given for  $\tau^0$ ,  $\sigma^0$ , and  $\gamma$  below was observed to be less sensitive regarding the image size and/or regularization parameter  $\alpha$  such that a high accuracy can be always expected after  $N$  iterations. Having  $u_\alpha^*$ , each algorithm was applied until the relative error  $\|u_\alpha^k - u_\alpha^*\|/\|u_\alpha^*\|$  is below a pre-defined threshold  $\epsilon$ . All algorithms were implemented in MATLAB and executed on a machine with 4 CPU cores, each 2.83 GHz, running a 64 Bit Linux system and MATLAB R2014a. Due to the built-in multi-threaded computation of diverse functions and expressions [24], MATLAB sessions were started with the option 'singleCompThread' limiting the computation to a single thread.

Three different images, each of them in sizes  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$ , were used to evaluate the performance of algorithms (see Fig. 1). All images were disturbed by Poisson statistics which was generated by `random('Poisson', 50  $\bar{u}$ , size( $\bar{u}$ )) / 50` with  $\bar{u}$  being the original image scaled to  $[0, 1]$ . Then, (1) was performed with  $w = f$  [13] and Fig. 2 exemplary shows the denoising capability on one of the images used in our experiments. For the performance evaluation, the following algorithms and parameter settings were used:

- **FGP**: Accelerated projected (sub)gradient descent algorithm (*Algorithm 1*).
- **(Fast)SB**: Split Bregman algorithm (*Algorithm 2*), respectively its accelerated version. Two Jacobi iterations were used to solve approximately the step ( $i$ ). We also tested the preconditioned conjugate gradient (PCG) method with warm starting and two different preconditioners (diagonal and circular) [25]. In case of circular preconditioner the matrix inversion was performed via the discrete cosine transform [12] using MATLAB's function `dctmtx`. Two parameter values of  $\mu_v$  were applied, an experimentally determined value  $\mu_v = 10^{-4}$  and  $\mu_v = \text{median}\{w^{-1}\}/\|-\Delta\|$  [26], where the later setting is motivated by improving the condition number of matrix  $I - \mu_v w \Delta$  in ( $i$ ).
- **(Fast)ADMM**: ADMM method (*Algorithm 3*), respectively its accelerated version. The following parameter settings were applied: (1)  $\mu_v = 10^{-4}$  and  $\mu_{\bar{u}} = 1$  (experimental); (2)  $\mu_v = \mu_{\bar{u}}/\|-\Delta\|$  and  $\mu_{\bar{u}} = \text{median}\{w^{-1}\}$  (motivated by [26]). The linear equation in substep ( $i$ ) was solved by the discrete cosine transform [12] using MATLAB's function `dctmtx`.
- **FastCP**: Accelerated Chambolle-Pock primal-dual method (*Algorithm 4*). The parameters were set as  $\tau^0 = \sigma^0 = 1/\|\text{div}\|$  and  $\gamma = 0.35(\alpha\|w\|_\infty)^{-1}$  [6].
- **AHMOD**: Modified Arrow-Hurwicz primal-dual method which is *Algorithm 4* with  $\bar{u}^{k+1} = u^{k+1}$  in ( $iv$ ). The parameters were chosen as following:  $\tau^0 = 0.02$ ,  $\sigma^0 = 4(\tau^0\|\text{div}\|^2)^{-1}$ , and  $\gamma = 0.35(\alpha\|w\|_\infty)^{-1}$  [6].

Tables 1 and 2 show representatively the results of performance evaluation for the images of size  $512 \times 512$  with  $\alpha = 10^{-4}$  and  $\alpha = 10^{-3}$ , respectively. As expected, the computation of solution will get harder with stronger regularization (i.e., larger regularization parameter  $\alpha$ ). In general, the modified Arrow-Hurwicz algorithm appears to be the most efficient method, not only regarding the CPU time but also with respect to the number of iterations in case of higher approximation accuracy (i.e., smaller  $\epsilon$ ). The computation time of FastCP algorithm



**Fig. 1.** Test images used to evaluate the performance of algorithms. Each image was evaluated in three different image sizes ( $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$ ).



**Fig. 2.** Image denoising using the "weighted" ROF model (1) with  $w = f$  [13]. From left to right: Poisson perturbed image of size  $512 \times 512$ , denoised reference image  $u_\alpha^*$  for  $\alpha = 10^{-4}$  and  $\alpha = 10^{-3}$ , respectively

is approximately twice the one of AHMOD but still outperforms the rest of the algorithms in the most cases. In addition, the following observations were made during the performance evaluation:

- In case of SB and ADMM based variants, the experimental determined parameters  $\mu_v = 10^{-4}$  and  $\mu_{\tilde{u}} = 1$  perform better for stronger regularization (i.e., larger  $\alpha$ ), whereby  $\mu_v = \mu_{\tilde{u}} / \|\Delta\|$  and  $\mu_{\tilde{u}} = \text{median}\{w^{-1}\}$  (motivated by [26]) provide a faster convergence for weaker regularization.
- In case of SB algorithm, the Jacobi iteration to solve (i) in *Algorithm 2* provided a better performance regarding the CPU time and PCG with circular preconditioner regarding the number of iterations. However, circular PCG was clearly slower caused by solving (multiple) circular linear equations in each PCG loop. In our experiments, one or two PCG iterations were sufficient to decrease the approximation error rapidly and diagonal preconditioner provided a balance between CPU time and number of iterations.

Finally, to get an impression regarding the behavior of algorithms in dependency on image size, the mean CPU time is plotted in Fig. 3. The plots show that

**Table 1.** Performance evaluation using the images of size  $512^2$  shown in Fig. 1. The table displays the CPU time in seconds and the number of iterations (both as mean and standard deviation over the images in Fig. 1) which were required to get the relative error of the solution below the error tolerance  $\epsilon$  for  $\alpha = 10^{-4}$ . For reasons of clarity, we show only the best results regarding the CPU time and number of iterations from each bunch of combinations discussed in case of SB and ADMM above.

$\alpha = 10^{-4}$	$\epsilon = 10^{-3}$				$\epsilon = 10^{-5}$			
	time		its		time		its	
	mean	std	mean	std	mean	std	mean	std
FGP	0.89	(0.02)	20	(1)	6.85	(0.38)	154	(14)
SB (best time)	1.19	(0.05)	<b>12</b>	<b>(1)</b>	21.74	(15.16)	224	(164)
SB (best its)	1.19	(0.05)	<b>12</b>	<b>(1)</b>	55.08	(45.75)	204	(165)
ADMM	3.17	(0.12)	15	(1)	51.53	(16.35)	242	(76)
FastCP	0.71	(0.02)	22	(1)	2.96	(0.35)	91	(11)
AHMOD	<b>0.40</b>	<b>(0.02)</b>	13	(1)	<b>1.52</b>	<b>(0.17)</b>	<b>48</b>	<b>(5)</b>

**Table 2.** Performance for images of size  $512^2$  and  $\alpha = 10^{-3}$  (see Table 1 for details)

$\alpha = 10^{-3}$	$\epsilon = 10^{-3}$				$\epsilon = 10^{-5}$			
	time		its		time		its	
	mean	std	mean	std	mean	std	mean	std
FGP	18.18	(4.61)	406	(103)	138.33	(25.96)	3085	(587)
SB (best time)	28.87	(5.20)	422	(81)	187.65	(48.32)	1357	(329)
SB (best its)	49.51	(12.48)	126	(31)	187.65	(48.32)	1357	(329)
ADMM	15.64	(3.24)	<b>74</b>	<b>(15)</b>	388.06	(207.87)	2456	(1321)
FastCP	10.74	(2.92)	335	(91)	59.99	(15.71)	1865	(486)
AHMOD	<b>5.36</b>	<b>(1.40)</b>	173	(45)	<b>29.18</b>	<b>(7.55)</b>	<b>937</b>	<b>(243)</b>

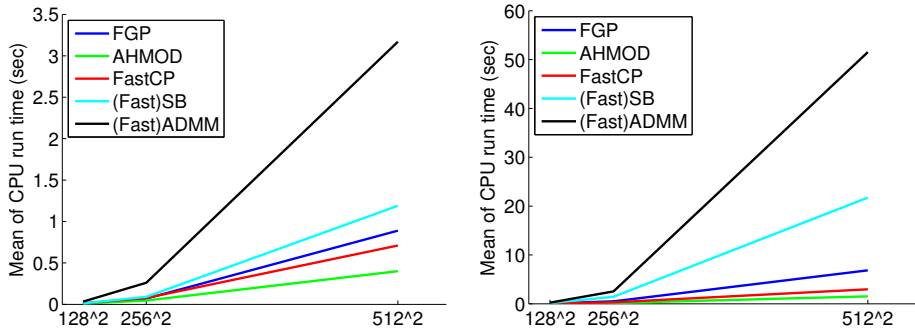
AHMOD, FastCP, and FGP seem to be more stable with respect to the change of the image size.

## 4 Discussion

The choice of  $\mu_v$  and  $\mu_{\bar{u}}$  in *Algorithms 2* and *3* influences the speed of convergence. However, since no a-priori knowledge exist how to set these parameters properly, variable parameters can be used to mitigate the performance dependency on the initial chosen fixed parameters. A strategy similar to [5], but with residuals defined in (3), was applied in our numerical experiments. With this strategy, the performance of ADMM based algorithms was improved in some cases but still insufficient to be competitive to the AHMOD algorithm such that the approach was not discussed in this manuscript more precisely.

The presented performance results are valid only as long as the weight  $z$  in the weighted least-squares data term  $\|\cdot\|_z^2$  is strictly positive. In [15], the fidelity





**Fig. 3.** CPU time of algorithms in seconds required to get the relative error of the solution below the error tolerance  $\epsilon = 10^{-3}$  (left) and  $\epsilon = 10^{-5}$  (right) for  $\alpha = 10^{-4}$  as a function of image size. Displayed are the mean values of time over the images in Fig. 1 and only the fastest variants in case of SB and ADMM algorithms are presented.

weight can also include a binary image such that the positivity condition on  $z$  is violated. In this case, the fidelity term is not longer uniformly convex such that the acceleration strategy included in *Algorithm 4* cannot be exploited anymore and ADMM based methods might present a better numerical performance.

**Acknowledgments.** The author thanks Martin Burger (University of Münster, Germany) for valuable comments that improved the manuscript. AS was supported in part by the German Federal Ministry of Education and Research BMBF through the project HYPERMATH and the German Research Foundation DFG via SFB 656.

## References

1. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* 60, 259–268 (1992)
2. Parikh, N., Boyd, S.: Proximal algorithms. *Found. Trend. Optim.* 1(3), 127–239 (2013)
3. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* 18(11), 2419–2434 (2009)
4. Setzer, S., Steidl, G., Morigenthaler, J.: A cyclic projected gradient method. *Comput. Optim. Appl.* 54(2), 417–440 (2013)
5. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trend. Mach. Learn.* 3(1), 1–122 (2011)
6. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* 40(1), 120–145 (2011)
7. Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci.* 3(4), 1015–1046 (2010)

8. Seybold, T., Keimel, C., Knopp, M., Stechele, W.: Towards an evaluation of denoising algorithms with respect to realistic camera noise. In: Proc. IEEE Int. Symp. Multimedia, pp. 203–210 (2013)
9. Hai Thai, T., Cogramne, R., Retraint, F.: Camera model identification based on the heteroscedastic noise model. *IEEE Trans. Image Process.* 23(1), 250–263 (2014)
10. Loupas, T., McDicken, W.N., Allan, P.L.: An adaptive weighted median filter for speckle suppression in medical ultrasonic images. *IEEE Trans. Circuits Syst.* 36(1), 129–135 (1989)
11. Jin, Z., Yang, X.: A variational model to remove the multiplicative noise in ultrasound images. *J. Math. Imaging Vis.* 39(1), 62–74 (2011)
12. Sawatzky, A., Brune, C., Kösters, T., Wübbeling, F., Burger, M.: EM-TV Methods for Inverse Problems with Poisson Noise. In: Level Set and PDE Based Reconstruction Methods in Imaging, pp. 71–142. Springer (2013)
13. Sawatzky, A., Brune, C., Müller, J., Burger, M.: Total variation processing of images with Poisson statistics. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 533–540. Springer, Heidelberg (2009)
14. Chouzenoux, E., Pesquet, J.-C., Repetti, A.: Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *J. Optim. Theory Appl.* (2013)
15. Sawatzky, A., Tenbrinck, D., Jiang, X., Burger, M.: A variational framework for region-based segmentation incorporating physical noise models. *J. Math. Imaging Vis.* 47(3), 179–209 (2013)
16. Green, P.J.: Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *J. R. Stat. Soc. B Met.* 46(2), 149–192 (1984)
17. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 136–152. Springer, Heidelberg (2005)
18. Beck, A., Teboulle, M.: A fast shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.* 2(1), 183–202 (2009)
19. Nemirovsky, A.S., Yudin, D.B.: Problem Complexity and Method Efficiency in Optimization. Wiley (1983)
20. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publisher (2004)
21. Goldstein, T., Osher, S.: The split Bregman method for  $L^1$ -regularized problems. *SIAM J. Imaging Sci.* 2(2), 323–343 (2009)
22. Goldstein, T., O’Donoghue, B., Setzer, S.: Fast alternating direction optimization methods. Technical Report CAM 12-35, UCLA (2012)
23. He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. *SIAM J. Imag. Sci.* 5(1), 119–149 (2012)
24. MATLAB Answers, <http://www.mathworks.com/matlabcentral/answers/95958>
25. Fessler, J.A., Booth, S.D.: Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction. *IEEE Trans. Image Process.* 8, 688–699 (1999)
26. Ramani, S., Fessler, J.A.: Convergent iterative CT reconstruction with sparsity-based regularization. In: Proc. 11th Int. Mtg., FULLY3D 2011, pp. 302–305 (2011)