Elsevier Editorial System(tm) for Communications in Nonlinear Science and Numerical Simulation
Manuscript Draft

Manuscript Number:

Title: Limited-memory scaled gradient projection methods for real-time image deconvolution in microscopy

Article Type: SI: CNSNS NUMTA2013

Section/Category: 5. Computational Methods and Simulations {S. Ruffo}

Keywords: image deconvolution; constrained optimization; scaled gradient projection method; Ritz values; GPU

Corresponding Author: Dr. Riccardo Zanella,

Corresponding Author's Institution: University of Ferrara

First Author: Riccardo Zanella

Order of Authors: Riccardo Zanella; Federica Porta; Gaetano Zanghirati; Luca Zanni

Abstract: Gradient projection methods have given rise to effective tools for image deconvolution in several relevant areas, such as microscopy, medical imaging and astronomy. Due to the large scale of the optimization problems arising in nowadays imaging applications and to the growing request of real-time reconstructions, an interesting challenge to be faced consists in designing new acceleration techniques for the gradient schemes, able to preserve the simplicity and low computational cost of each iteration. In this work we
propose an acceleration strategy for a state of the art scaled gradient projection method for image deconvolution in microscopy. The acceleration idea is derived by adapting a step-length selection rule, recently introduced for limited-memory steepest descent methods in unconstrained
optimization, to the special constrained optimization framework arising in image reconstruction. We describe how important issues related to the generalization of the step-length rule to the imaging optimization problem
have been faced and we evaluate the improvements due to the acceleration strategy by numerical experiments on large-scale image deconvolution problems.

Limited-memory scaled gradient projection methods for real-time image deconvolution in microscopy[☆]

F. Porta[a,1], R. Zanella[b,3,*], G. Zanghirati[b,2], L. Zanni[a]

[a] *Department of Physics, Computer Science and Mathematics, University of Modena and Reggio Emilia, Italy*
[b] *Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy*

Abstract

Gradient projection methods have given rise to effective tools for image deconvolution in several relevant areas, such as microscopy, medical imaging and astronomy. Due to the large scale of the optimization problems arising in nowadays imaging applications and to the growing request of real-time reconstructions, an interesting challenge to be faced consists in designing new acceleration techniques for the gradient schemes, able to preserve the simplicity and low computational cost of each iteration. In this work we propose an acceleration strategy for a state of the art scaled gradient projection method for image deconvolution in microscopy. The acceleration idea is derived by adapting a step-length selection rule, recently introduced for limited-memory steepest descent methods in unconstrained optimization, to

[☆] This paper is submitted to the NUMTA2013 Conference Proceedings

[*] Corresponding author

Email addresses: `federica.porta@unimore.it` (F. Porta), `r.zanella@unife.it` (R. Zanella), `g.zanghirati@unife.it` (G. Zanghirati), `luca.zanni@unimore.it` (L. Zanni)

[1] Partially supported by the Italian Spinner2013 PhD Project *"High-complexity inverse problems in biomedical applications and social systems"*.

[2] Partially supported by the local research projects "FAR2011 – *NOCSiMA*", "FAR2012 – *Optimization Methods for Inverse Problems*" of the University of Ferrara.

[3] Partially supported by the FIRB2012 project of the Italian Ministry for University and Research (grant n. RBFR12M3AC) and by the local research projects "FAR2011 – *NOCSiMA*", "FAR2012 – *Optimization Methods for Inverse Problems*" of the University of Ferrara

All the authors are partially supported by the "GNCS-INdAM".

Preprint submitted to Communications in Nonlinear Science and Numerical Simulation December 18, 2013

the special constrained optimization framework arising in image reconstruction. We describe how important issues related to the generalization of the step-length rule to the imaging optimization problem have been faced and we evaluate the improvements due to the acceleration strategy by numerical experiments on large-scale image deconvolution problems.

Keywords: image deconvolution, constrained optimization, scaled gradient projection method, Ritz values, GPU
2010 MSC: 65K05, 68U10, 65Y10, 65Y05

1. Introduction

Image deconvolution is a useful technique for improving the image quality of many types of microscope. Unfortunately, in case of large-scale imaging problems, the deconvolution can require a too large computational time which leads to undesirable delay in the reconstruction process. This is mainly due to the slow convergence rate of the iterative deconvolution methods usually exploited on microscopy images, such as the well-known scaled gradient minimization method called Richardson-Lucy (RL) algorithm [1, 2]. To overcome this disadvantage, two strategies have been exploited in the last years: the acceleration of the deconvolution algorithms and the implementation of the algorithms on multiprocessor architectures, like the Graphics Processing Units (GPU). The combination of the benefits from these strategies allowed to achieve promising time reductions in the deconvolution process, providing stimulus for further research in these fields. In this paper we focus on an iterative deconvolution algorithm that aims at exploiting a new step-length selection strategy for gradient descent methods to improve the convergence rate of state of the art deconvolution approaches in microscopy. The considered step-length strategy has been recently proposed by R. Fletcher [3] in the context of limited-memory steepest descent methods for unconstrained minimization problems. In [3] numerical evidence has been also provided indicating remarkable gain in the convergence rate over the classical Barzilai-Borwein (BB) step-length rule [4]. Since in the last years promising image reconstruction algorithms have been designed by exploiting BB-based rules within gradient methods [5, 6, 7, 8, 9, 10, 11], it is worthwhile to investigate if useful acceleration can be achieved with the new step-length selection idea. In particular, we focus on the algorithm for image deconvolution in microscopy provided by the Scaled Gradient Projection (SGP) method recently developed in [12], that can be appropriately modified for managing the step-length rule proposed in [3]. SGP is a very general algorithm for minimization problems with simple constraints, able to exploit both scaled gradient directions and selection rules for the step-length parameter. By combining an efficient step-length selection based on the adaptive alternation of the two BB rules [13, 14] and a scaling strategy similar to that used by the RL algorithm, SGP has shown significant convergence rate improvements in comparison with RL, without excessive growth of the cost per iteration and loss of reconstruction accuracy [5]. These abilities allowed the GPU version of SGP designed in [12] to show interesting performance as real-time deconvolution algorithm. In order to efficiently equip SGP with the new step-length rule presented in [3], crucial aspects concerning with the presence of scaled gradient directions and nonnegativity constraints need to be discussed. In this work we propose how to generalize the step-length rule to the SGP framework and provide numerical evidence of the gain achievable in comparison with the BB-based step-length selection previously exploited by SGP.

The paper is organized as follows. In section 2, the optimization problem arising in image deconvolution is stated and the RL and SGP iterative regularization algorithms are recalled; in section 3, the new step-length rule for SGP is introduced and, in section 4, a computational study is presented for validating on large-scale image deconvolution test problems the SGP version equipped with the new rule. Finally, some conclusions and a discussion on possible future work are reported in section 5.

2. The SGP method for image deconvolution

We shortly introduce the optimization problem arising from the maximum likelihood approach to image deconvolution in microscopy; for a deeper discussion of the image deconvolution problem, we refer the reader to [15, 16]. Let us denote by $\boldsymbol{y} \in \mathbb{R}^n$ the detected blurred and noisy image and assume a linear model for describing the image acquisition process: $A\boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is the unknown object, $\boldsymbol{b} \in \mathbb{R}^n$ indicates the known positive background emission ($b_i > 0$) and A is the $n \times n$ imaging matrix representing the blurring phenomenon. The detected values y_i are nonnegative and the matrix A can be considered with nonnegative entries, generally dense and such that $\sum_j A_{ij} > 0 \ \forall i$ and $A^T\boldsymbol{e} = \boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{R}^n$ is a vector whose components are all equal to one. Furthermore, we may assume that periodic boundary conditions are imposed for the discretization of the Fredholm integral equation that models the image formation process, so that the matrix A is block-circulant with circulant blocks and the matrix-vector products $A\boldsymbol{x}$ can be done quickly, with $\mathcal{O}(n \log n)$ complexity, by using the Fast Fourier Transform (FFT) [17]. If we assume that the detected values y_i are realization of independent Poisson random variables, with unknown expected values $(A\boldsymbol{x} + \boldsymbol{b})_i$, the Maximum-Likelihood (ML) approach to the deconvolution problem leads to the minimization of a suitable data-fidelity function called *generalized Kullback-Leibler* (KL) *divergence* (or *Csiszár I-divergence*) [18, 19]:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \left\{ y_i \ln \frac{y_i}{(A\boldsymbol{x} + \boldsymbol{b})_i} + (A\boldsymbol{x} + \boldsymbol{b})_i - y_i \right\} \ , \tag{1}$$

whose gradient and hessian are given by

$$\nabla f(\boldsymbol{x}) = A^T \boldsymbol{e} - A^T Z^{-1} \boldsymbol{y} \tag{2}$$

$$\nabla^2 f(\boldsymbol{x}) = A^T Y Z^{-2} A, \tag{3}$$

where $Z = \mathrm{diag}(A\boldsymbol{x} + \boldsymbol{b})$ is a diagonal matrix with the entries of $(A\boldsymbol{x} + \boldsymbol{b})$ on the main diagonal and $Y = \mathrm{diag}(\boldsymbol{y})$. We observe that the hessian matrix is positive semidefinite in any point of the nonnegative orthant. Due to the ill-posedness of the image restoration problem [16], the matrix A could be very ill-conditioned and a solution of the convex optimization problem

$$\min_{\boldsymbol{x} \geq \boldsymbol{0}} \ f(\boldsymbol{x}) \tag{4}$$

does not provide sensible reconstructions of the unknown image. Two alternative strategies can be exploited to overcome this drawback. The first approach consists in looking for suited regularized reconstructions by early stopping iterative minimization methods applied to the problem (4). The second strategy requires to solve a regularized minimization problem

$$\min_{\boldsymbol{x} \geq \boldsymbol{0}} \ f(\boldsymbol{x}) + \beta f_R(\boldsymbol{x}), \tag{5}$$

where a regularization functional $f_R(\boldsymbol{x})$ is used for forcing some *a priori* information on the unknown image and the parameter $\beta > 0$ is used to control the trade-off between the data-fidelity term $f(\boldsymbol{x})$ and the regularization term. In this work we focus on the former approach, that in case of large-scale imaging problems is usually the preferred choice: it allows to check the reconstruction quality during the iterative process, without requiring expensive computation to set a suitable regularization parameter. Within the microscopy community, the most famous iterative regularization method for the nonnegative minimization of the KL divergence is the *Richardson-Lucy* (RL) algorithm [1, 2] (often called *Expectation Maximization* (EM) algorithm), whose iteration can be stated as follows:

$$\boldsymbol{x}^{(k+1)} = X_k A^T Z_k^{-1} \boldsymbol{y} = \boldsymbol{x}^{(k)} - X_k \nabla f(\boldsymbol{x}^{(k)}), \qquad \boldsymbol{x}^{(0)} > \boldsymbol{0}, \tag{6}$$

where $X_k = \mathrm{diag}(\boldsymbol{x}^{(k)})$ and $Z_k = \mathrm{diag}(A\boldsymbol{x}^{(k)} + \boldsymbol{b})$. As can be observed from equation (6), the RL algorithm is a special scaled gradient method where the diagonal scaling matrix has the current iteration on the main diagonal and the variables' nonnegativity is ensured by the assumptions on A, $\boldsymbol{b}$, $\boldsymbol{y}$ and $\boldsymbol{x}^{(0)}$. Convergence properties of the algorithm have been proved in various situations in [20, 21, 22, 23, 24, 25, 26]. The simple and inexpensive form of the iteration makes the algorithm very attractive, but it does not yields a fast convergence: in general, several hundreds (or thousands) of iterations are needed to obtain a suitably regularized reconstruction of the unknown image. The slow convergence seriously limits the use of the RL algorithm, especially in case of large-scale imaging problems, and the design of accelerated iterative schemes have received growing interest in the last years. For microscopy images deconvolution, a very promising accelerated reconstruction approach has been recently proposed in [12], based on the general constrained minimization scheme called *Scaled Gradient Projection* (SGP) method introduced in [5]. The SGP scheme is a gradient method able to exploit essentially four key elements: the scaled gradient directions, a parameter for controlling the step-length along the scaled gradient directions, a projection step for generating feasible descent directions and a line search strategy for ensuring sufficient reduction of the objective function during the iterations. All together, these elements make SGP a very flexible tool that, on one hand, recovers standard constrained minimization approaches and, on the other hand, can improve them by including most effective settings currently available in literature for those key elements. To better explain how SGP improves the RL algorithm, we shortly recall its iteration for problem (4). The reader is referred to [5] for more technical details and the convergence analysis of the general scheme. Given an initial feasible $\boldsymbol{x}^{(0)}$,

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{z}^{(k)}; \quad \boldsymbol{z}^{(k)} = \mathbb{P}_+(\boldsymbol{x}^{(k)} - \alpha_k D_k \boldsymbol{g}^{(k)}) - \boldsymbol{x}^{(k)} \tag{7}$$

where

- $D_k = \mathrm{diag}\big(d_1^{(k)}, \ldots, d_n^{(k)}\big)$ is the diagonal scaling matrix satisfying

$$\frac{1}{L} \leq d_i^{(k)} \leq L, \quad i = 1, \ldots, n, \quad \forall\, k, \qquad L \geq 1 \ ;$$

- α_k is the step-length parameter that must satisfy

$$0 < \alpha_{min} \leq \alpha_k \leq \alpha_{max}, \qquad \forall\, k \ ;$$

- $\mathbb{P}_+(\cdot)$ denotes the projection of a vector onto the nonnegative orthant, used in (7) to obtain the feasible descent direction $\boldsymbol{z}^{(k)}$;
- $\lambda_k \in [0, 1]$ is the line search parameter used to reduce, if necessary, the size of the step along the feasible descent direction $\boldsymbol{z}^{(k)}$, in order to guarantee a suitable improvement in the objective function; λ_k can be computed by means of a line search technique, such as the standard monotone Armijo rule [27] or the nonmonotone strategy introduced in [28].

The main feature of the above iterative scheme is the weakness of the conditions on both the diagonal scaling entries and the step-length parameter: they must be simply bounded below and above by positive constants. This leaves the user free to devise updating rules for these parameters, to achieve some specific goal (generally, the convergence rate improvement), while the line search step controlling the descent of the objective function ensures that every limit point of the sequence generated by SGP is a constrained stationary point of (4). When SGP is exploited for the nonnegative minimization of the KL divergence, effective selection for the scaling matrix and the step-length parameter have been proposed in [5]. For the scaling matrix, an updating rule similar to that exploited by the RL algorithm is suggested:

$$d_i^{(k)} = \mathrm{diag}\left(\min\left\{L, \max\left\{\frac{1}{L}, x_i^{(k)}\right\}\right\}\right), \qquad i = 1, \ldots, n \ ; \tag{8}$$

while, for the step-length parameter, a selection strategy derived by an updating rule introduced in [14] is used (see also [13]). This step-length selection, described in great detail in [5], is based on an adaptive alternation of the two Barzilai-Borwein (BB) rules [4], which in case of scaled gradient directions are defined by

$$\alpha_k^{\mathrm{BB1}} = \frac{\boldsymbol{s}^{(k-1)^T} D_k^{-2} \boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T} D_k^{-1} \boldsymbol{w}^{(k-1)}} \qquad \text{and} \qquad \alpha_k^{\mathrm{BB2}} = \frac{\boldsymbol{s}^{(k-1)^T} D_k \boldsymbol{w}^{(k-1)}}{\boldsymbol{w}^{(k-1)^T} D_k^{2} \boldsymbol{w}^{(k-1)}}$$

where $\boldsymbol{s}^{(k-1)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$ and $\boldsymbol{w}^{(k-1)} = \nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^{(k-1)})$. With the above settings for the scaling matrix and the step-length parameter, SGP has been compared against the RL algorithm on many imaging problems, providing a remarkable convergence rate acceleration without losing accuracy in the reconstruction. In particular, in [12] a special SGP implementation for GPU devices has been successfully exploited, providing a step ahead toward real-time deconvolution of microscopy images. Thus, further improvements of the SGP performance can be a crucial key for developing more effective deconvolution tools. In the next section we introduce a new step-length selection strategy, that can be exploited within SGP to achieve meaningful convergence rate improvements compared with the BB-based updating rule currently implemented.

3. A limited-memory step-length selection rule for SGP

The new rule that we propose to update the parameter α_k in SGP is derived from an idea recently introduced by R. Fletcher [3] for the choice of the step-length in steepest descent methods for unconstrained minimization problems. The step-length selection presented in [3] aims at capturing second order information by exploiting a limited number of back gradients. Starting from the unconstrained quadratic case

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \frac{1}{2} \boldsymbol{x}^T A \boldsymbol{x} \tag{9}$$

where A is a symmetric positive definite matrix, a step-length selection for the steepest descent methods

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)}, \qquad \boldsymbol{g}^{(k)} = A\boldsymbol{x}^{(k)}, \tag{10}$$

is introduced by using special approximations of the eigenvalues of A, called Ritz values [29]. Given a positive integer m, on the k-th iteration, the Ritz values on which the step-length rule is based are the eigenvalues θ_i, $i = 1, \ldots, m$ of the $m \times m$ symmetric tridiagonal matrix

$$T = Q^T A Q$$

where $Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_m]$ is the $n \times m$ matrix whose columns $\boldsymbol{q}_i$ are the orthonormal vectors generated by the Lanczos iterative process [30], applied to A, with starting vector $\boldsymbol{q}_1 = \boldsymbol{g}^{(k-m)} / \|\boldsymbol{g}^{(k-m)}\|$. These m Ritz values, in decreasing order, are then used to define the step-lengths for the next m iterations:

$$\alpha_{k+i-1} = \theta_i^{-1}, \qquad i = 1, \dots, m\,. \tag{11}$$

To extend this idea to general non-quadratic functions and to make the selection strategy more convenient from the computational point of view, important remarks are reported in [3] that allow one to rewrite the matrix T in terms of $\boldsymbol{g}^{(k)}$ and the most recent m back gradients,

$$G = \begin{bmatrix}\boldsymbol{g}^{(k-m)} & \dots & \boldsymbol{g}^{(k-2)} & \boldsymbol{g}^{(k-1)}\end{bmatrix}. \tag{12}$$

without using Q and A. In fact, from (10) it follows that the columns of G are in the Krylov sequence initiated by $\boldsymbol{g}^{(k-m)}$

$$\left\{\boldsymbol{g}^{(k-m)},\ A\boldsymbol{g}^{(k-m)},\ A^2\boldsymbol{g}^{(k-m)},\ \dots\ ,\ A^{m-1}\boldsymbol{g}^{(k-m)}\right\}\ ,$$

and, taking into account that the columns of Q are special orthonormal basis vectors for the Krylov sequence, we may rewrite G as

$$G = QR$$

where R is upper triangular and nonsingular, if the columns of G are linearly independent. This means that we have

$$T = Q^T A Q = R^{-T} G^T A G R^{-1}$$

and since $AG = \begin{bmatrix}G & \boldsymbol{g}^{(k)}\end{bmatrix} J$, where

$$J = \begin{bmatrix} \alpha_{k-m}^{-1} & & \\ -\alpha_{k-m}^{-1} & \ddots & \\ & \ddots & \alpha_{k-1}^{-1} \\ & & -\alpha_{k-1}^{-1} \end{bmatrix},$$

we obtain

$$T = R^{-T}\begin{bmatrix}G^T G & G^T\boldsymbol{g}^{(k)}\end{bmatrix} J R^{-1} = R^{-T}\begin{bmatrix}R^T R & G^T\boldsymbol{g}^{(k)}\end{bmatrix} J R^{-1}.$$

Finally, by denoting with $\boldsymbol{r}$ the vector satisfying $R^T\boldsymbol{r} = G^T\boldsymbol{g}^{(k)}$, the matrix T can be rewritten as

$$T = \begin{bmatrix}R & \boldsymbol{r}\end{bmatrix} J R^{-1}. \tag{13}$$

Equation (13) represents a clever and efficient way for generalizing the step-length selection (11) based on the Ritz values to the non-quadratic case, requiring essentially only an extra area of memory for the $n \times m$ matrix G and the computation of $G^T G$ for obtaining R via a Cholesky factorization. The number m of back gradients is usually a very small positive integer due to practical limits such as the memory requirements and the round-off error in the computation of R. In the computational study reported in [3], the value $m = 5$ is suggested and in the experiments described below we found convenient even smaller values: $m = 3$ or $m = 4$. Thus, in case of large-scale problem, the operations whose computational cost depends on m can be considered inexpensive.

To exploit this step-length selection within the SGP scheme (7) for the constrained optimization problem (4), some important questions must be addressed, in particular concerning with the presence of the non-quadratic objective function, the scaling matrix multiplying the gradient direction and the nonnegative constraints.

In the general non-quadratic case the matrix T has an upper Hessenberg structure and, following the suggestion in [3], we construct a symmetric tridiagonal matrix $\tilde{T}$ by substituting the strict upper triangle of T with the transpose of its strict lower triangle; in this way, the eigenvalues of $\tilde{T}$ works as Ritz-like values and reduce to the Ritz values in the quadratic case. Furthermore, the matrix $\tilde{T}$ can have some negative eigenvalues that we keep out from the set of available Ritz-like values, by providing SGP with $m_1 < m$ step-lengths for the next iterations. After these iterations, a set of m_1 gradient vectors are obtained that at first are exploited for defining step-lengths by means of an $n \times m_1$ matrix G, and then will be used as back gradients for filling a new matrix G up to m columns.

The presence of the diagonal scaling matrix can be taken into account by observing its effect in the simple unconstrained quadratic case (9). In this case, a scaled gradient step $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k D_k \boldsymbol{g}^{(k)}$, can be viewed as a non-scaled step on a transformed problem obtained with the linear transformation of variables $\boldsymbol{y} = D_k^{-1/2}\boldsymbol{x}$. In fact, the transformed objective function becomes $\tilde{f}(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^T D_k^{1/2} A D_k^{1/2} \boldsymbol{y}$ and for the transformed gradient vector we have $\tilde{\boldsymbol{g}} = D_k^{1/2} A D_k^{1/2} \boldsymbol{y}$; then, by assuming $\boldsymbol{y}^{(k)} = D_k^{-1/2}\boldsymbol{x}^{(k)}$, we can write

$$\begin{aligned}
\boldsymbol{y}^{(k+1)} &= \boldsymbol{y}^{(k)} - \alpha_k \tilde{\boldsymbol{g}}^{(k)} = D_k^{1/2}\boldsymbol{x}^{(k)} - \alpha_k D_k^{1/2} A D_k^{1/2} \boldsymbol{y}^{(k)} \\
&= D_k^{-1/2}\left(\boldsymbol{x}^{(k)} - \alpha_k D_k \boldsymbol{g}^{(k)}\right) = D_k^{-1/2}\boldsymbol{x}^{(k+1)}.
\end{aligned}$$

Thus, a natural way to extend the above step-length selection procedure to the case of diagonally scaled gradient methods consists in defining the step-lengths in the transformed variables, by storing in G the transformed gradient vectors $D_{k-i}^{1/2}\nabla f(\boldsymbol{x}^{(k-i)})\,,\ i = 1,\ldots,m\,$, and by using $D_k^{1/2}\nabla f(\boldsymbol{x}^{(k)})$ to define the vector $\boldsymbol{r}$ in (13). A similar approach is suggested in [3], by looking at the scaling matrix as arising from the application of preconditioning techniques to the problem (9).

Finally, we extend the step-length selection rule to the nonnegatively constrained case by observing that in the solution $\boldsymbol{x}^*$ of (4) the components of the gradient corresponding to the free variables ($x_i^* > 0$) must be zero. This means that, during the minimization iterative process, those components must be reduced to zero (in a fashion similar to the unconstrained case) and then we try to define the step-lengths by giving more importance to those components of the gradient. We follow this idea by substituting in the step-length rule the gradients of the objective function with the vectors

$$\nabla_{\rm nz} f(\boldsymbol{x}^{(k-i)}) = \begin{cases} 0 & \text{if} \quad x_j^{(k-i)} = 0 \\ \left(\nabla f(\boldsymbol{x}^{(k-i)})\right)_j & \text{if} \quad x_j^{(k-i)} > 0 \end{cases} \qquad i = 0, 1, \ldots, m\,.$$

In short, the matrix G that we exploit to define the Ritz-like values within SGP is the following:

$$G = \left[D_{k-m}^{1/2}\nabla_{\rm nz} f(\boldsymbol{x}^{(k-m)}) \quad \ldots \quad D_{k-2}^{1/2}\nabla_{\rm nz} f(\boldsymbol{x}^{(k-2)}) \quad D_{k-1}^{1/2}\nabla_{\rm nz} f(\boldsymbol{x}^{(k-1)}) \right].$$

4. Numerical experiments

In this section we report the results of the numerical tests we performed to assess the effectiveness of the new SGP steplength selection rule on a meaningful application, namely the deconvolution of confocal microscopy images. Given that we are interested in evaluating the advantages of the algorithm in terms of iterations and computational time, while maintaining a sufficiently good reconstruction (that is to say, an image/volume reconstruction having an error level comparable to the one of other reference methods for the application field), we use synthetic phantoms. Even if they are randomly generated, these phantoms are properly designed to correctly simulate real-world situations and have already been tested and validated by experts in the field [12]. Seeking completeness, the next subsection summarizes the generation process. The subsection 4.2 describes the numerical behaviour of the SGP algorithm equipped with the rule based on the Ritz-like values, called SGP-Ritz, compared against the classical RL method (6) as well as the standard SGP algorithm, which has a step-length selection based on the adaptive alternation of the two BB rules described in [5, 12]. The aim of this experimentation is to highlight how the new limited-memory step-length approach presented in this work allows SGP to improve its performance in terms of convergence rate and, consequently, reconstruction time. Even if the reported results are still limited to a particular application, they provide a meaningful indication, which is also confirmed by additional experiences we performed (not reported here, because they go beyond the scope of this work). Finally, the interesting additional time reduction due to a suited GPU implementation of the iterative deconvolution algorithms is discussed in subsection 4.3.

4.1. Synthetic data sets

As it is well known, in addition to their theoretical properties, deconvolution algorithms have usually to be assessed also with respect to practical application fields where potentially they will be used. For this purpose, one should use test sets as close as possible to the realistic environment of those applications. This is a crucial issue to consider, particularly when algorithm's robustness is concerned.

In this paper we focus our attention on microscopy imaging applications: namely, on the reconstruction of micro-tubule network inside the cell. To this end, a routine has been implemented which is able to pseudo-randomly generate phantoms which mimic the micro-tubule cytoskeleton of a real cell. This routine randomly selects the starting positions of a given and fixed number of filaments. Then, a stochastic process is used for iteratively choosing the directions of filaments' growth. By selecting either a bi- or a three-dimensional growth one obtains 2D or 3D phantoms, respectively. Those filaments are assumed to have a tubular structure with radius of 30nm. To further improve the data realism, at each growth step of each filament we associate a value in the range $[0, 1]$, which accounts for the heterogeneity of protein concentration between different filaments. In the present case, three values are used: 0.5, 0.75 and 1. At the end of this generation process we get a clean 2D or 3D network structure that represents the cell's inner micro-tubule network, that is to say the *real object* one would like to recover.

The next step is to generate the *ideal object* (not noisy, blurred only): we convolve the phantom $\tilde{x}$ with the system's *point spread function* (PSF), *i.e.*,

$H * \tilde{\boldsymbol{x}}$. Clearly the PSF depends on the acquisition system technology. In our experiments we consider a confocal microscopy and for mimicking its spatial resolution we model the PSF by a radially symmetric Gaussian function [31]:

$$h_{\mathrm{CLSM}}(r, z) = \exp\bigl(-r^2/(2\sigma_r^2)\bigr) \exp\bigl(-z^2/(2\sigma_z^2)\bigr) \quad , \tag{14}$$

which reduces to $h_{CLSM}(r) = \exp\bigl(-r^2/(2\sigma_r^2)\bigr)$ in the 2D case. Here, σ is related to the full-width at half-maximum (FWHM) by $\mathrm{FWHM} = 2\sigma\sqrt{2\ln(2)}$. Estimates for both σ_r and σ_z in (14) can be recovered from real detections (see for instance [12]). In the present case we use $\mathrm{FWHM}_r = 220\mathrm{nm}$, $\sigma_r = 93\mathrm{nm}$ and, for the 3D cases, $\mathrm{FWHM}_z = 600\mathrm{nm}$, $\sigma_z = 255\mathrm{nm}$. After the convolution, to obtain the ideal object in terms of average detected photons, the convolved object is multiplied by a number $\tau > 0$ which depends on several factors (the emission rate of the fluorophore, the collection efficiency of the system and the pixel dwell-time). This completes the blurring process and gives the *ideal object*, that is what the acquisition system would get, if no noise existed.

However, a completely un-noisy detection is far from reality. Here a reasonable assumption is that photon counting noise represents the major source of noise for the detection process, together with a constant background $\boldsymbol{b}$ that further degrades the acquired observation. Thus, the final step consists in corrupting the i-th pixel/voxel with a Poisson process with mean $\tau(H * \tilde{\boldsymbol{x}})_i + b_i$. This in turn makes possible to control the signal-to-noise ratio (SNR): by increasing τ, the average number of detected photons increases and hence the SNR increases. More precisely, SNR and τ are linked as follows:

$$\mathrm{SNR}_{\mathrm{db}} = 10 \log \left(\max_i \frac{\tau(H * \boldsymbol{x})_i}{\sqrt{\tau(H * \boldsymbol{x})_i + b_i}} \right) \quad . \tag{15}$$

At the end of this process we get the phantom, that is what would be the *observed object*. Notice that, even if the background $\boldsymbol{b}$ can be different at different pixels/voxels, it is most often assumed uniformely constant, that is to say $b_i = b$, $\forall i$.

The degrading process just described provides us with sufficiently realistic phantoms to use as test sets for the algorithms deconvolution assessment. In particular, given that in simulation the object $\tilde{\boldsymbol{x}}$ used to generate the observed object is known, during algorithms execution we are able to quantitatively monitor at each k-th iteration the quality of the deconvolved objects.

Moreover, it is worth noticing that we deconvolved the synthetic objects by means of the very same PSFs used for their generation (it is the so-called *inverse crime*). Thereby no fundamental biased affects the results due to the choice of the PSF.

4.2. Numerical results

For a first intensive validation of SGP-Ritz in comparison with RL and the standard SGP, all the methods are implemented in Matlab 8.1.0 and the experiments are performed on a computer equipped with an Intel Xeon X5690 SixCore CPUs at 3.47GHz. As concerning the step-length parameters, we use the setting suggested in [5] for the standard SGP while in SGP-Ritz we fix $m = 3$. The parameters defining the monotone line search, chosen by both SGP and SGP-Ritz to determine λ_k, and the constant L, employed to bound the diagonal entries of D_k, have been set as described in the Supplementary Informations of [12].

To evaluate the performances of the discussed algorithms, we carried out several numerical experiments on synthetic microscopy test problems with different features, generated as described in the previous subsection. For the bi-dimensional case we considered square images of increasing size n, namely 128^2, 256^2, 512^2, 1024^2. For the three-dimensional case, we studied the algorithms performances on some volumes: a small ($128 \times 128 \times 64$ voxels), a medium ($256 \times 256 \times 32$ voxels) and a large ($512 \times 512 \times 64$ voxels) test set. For each one of those data sets we took into account two different noise values: the corresponding test problems are indicated with "low noise" and "high noise". Figure 1 shows the true object and the blurred and noisy data for the bi-dimensional test problems. We only report the corrupted images for the high noise level because the differences against the images affected by low noise are not remarkable from the visual point of view.

The methods' effectiveness in approximating the original image has been assessed through the relative reconstruction error, defined as $\|\boldsymbol{x}^{(k)} - \tilde{\boldsymbol{x}}\| \,/\, \|\tilde{\boldsymbol{x}}\|$, where $\tilde{\boldsymbol{x}}$ is the original image and $\boldsymbol{x}^{(k)}$ is the reconstruction after k iterations. Tables 1 and 2 report the minimum relative reconstruction error (*Opt. err.*), the number of iterations (*Opt. it.*) and the computational time in seconds (*Sec.*) required to get the minimum error for the bi-dimensional and the three-dimensional test problems, respectively. In Table 2, we marked with an asterisk the case where a method reached the prefixed maximum number of iterations without achieving the minimum reconstruction error and we reported the error and the computational time corresponding to that maximum number of iterations.

Table 1: Numbers of iterations, reconstruction time and minimum error reached in solving the 2D deblurring problems.

	high noise			low noise		
	Opt. it.	*Sec.*	*Opt. err.*	*Opt. it.*	*Sec.*	*Opt. err.*
			128 × 128			
RL	988	1.7	0.48	2280	3.8	0.46
SGP	180	0.6	0.46	377	1.1	0.45
SGP-Ritz	107	0.4	0.46	190	0.6	0.45
			256 × 256			
RL	1320	6.1	0.48	1689	8.0	0.47
SGP	206	1.9	0.46	209	2.0	0.45
SGP-Ritz	131	1.4	0.46	144	1.5	0.45
			512 × 512			
RL	1027	17.4	0.46	1548	26.7	0.45
SGP	165	5.1	0.44	210	6.4	0.42
SGP-Ritz	104	3.8	0.44	174	6.3	0.42
			1024 × 1024			
RL	1002	69.9	0.46	1481	100.9	0.45
SGP	113	15.3	0.44	163	19.6	0.43
SGP-Ritz	95	13.8	0.45	110	15.4	0.43

We can observe from Tables 1 and 2 that both the versions of the SGP method are able to reach a minimum reconstruction error lower than the one provided by the RL algorithm. Since for the 3D test problems the Matlab implementations require a long reconstruction time, we decide to also consider the performances of SGP and SGP-Ritz in simply achieving the RL minimum error. Table 3 shows the number of iterations (*It.*) and the computational time (*Sec.*) needed by SGP and SGP-Ritz to obtain the RL accuracy (*Err.*).

The main conclusion that can be drawn from these experiments is that the limited-memory updating rule for the step-length selection allows the scaled gradient projection method to reach better performances, in terms of computational time and number of iterations, than the original choice of α_k proposed in [5], while still preserving the same reconstruction accuracy. This is shown also by Figure 2, where the relative reconstruction error and the decrease of the objective function are plotted as a function of the number of iterations. Only two data sets are profiled as an example in the figure, be-

Table 2: Numbers of iterations, reconstruction time and minimum error reached in solving the 3D deblurring problems.

	high noise			low noise		
	Opt. it.	*Sec.*	*Opt. err.*	*Opt. it.*	*Sec.*	*Opt. err.*
			$128 \times 128 \times 64$			
RL	524	43.4	0.61	470	34.9	0.61
SGP	411	53.6	0.60	380	45.0	0.60
SGP-Ritz	207	31.4	0.60	218	30.1	0.60
			$256 \times 256 \times 64$			
RL	868	311.3	0.52	837	300.8	0.52
SGP	446	280.9	0.51	617	372.3	0.50
SGP-Ritz	294	208.0	0.51	329	229.1	0.50
			$512 \times 512 \times 64$			
RL	1568	2158.9	0.49	2500*	3400.0*	0.56*
SGP	750	1773.1	0.48	1319	3164.6	0.52
SGP-Ritz	372	1013.1	0.48	897	2438.7	0.52

cause an analogous behavior is observed in every other test problem. In short, SGP-Ritz exhibits an improved convergence rate, without degrading neither the simplicity, nor the low computational cost per iteration of the standard SGP. Furthermore, in comparison with the RL algorithm, a remarkable gain in the reconstruction time is achieved.

For illustrative purpose, in Figures 3 and 4 we show the bi-dimensional reconstructions of the 512 × 512 and the 1024 × 1024 images returned by the different methods. Figures 5–10 report the restoration of the transversal, sagittal and coronal central sections for two 3D data sets, provided by the three methods stopped at RL minimum error. It is useful to notice in Figure 8 how different is the observed transversal slice with respect to the one of the original object. This is exactly due to the blurring effect, which thicken the filaments and makes them overflow in adjacent layers. Thus, the deblurring algorithm is successful if it is able to remove that overflow to the maximum extent. The reconstruction panels show that the overflow is greatly reduced.

4.3. GPU implementation for real-time deconvolution

In this subsection we describe the hardware issues and the implementation strategies that allow to greatly accelerate the computations involved by the SGP schemes described above, thus making large-scale multidimen-

Table 3: Numbers of iterations, reconstruction time and error reached in solving the 3D deblurring problems with the RL-accuracy.

	high noise			low noise		
	It.	*Sec.*	*Err.*	*It.*	*Sec.*	*Err.*
128 × 128 × 64						
RL	524	43.4	0.61	470	34.9	0.61
SGP	323	41.2	0.61	177	22.7	0.61
SGP-Ritz	140	20.9	0.61	132	19.6	0.61
256 × 256 × 64						
RL	868	311.3	0.52	837	300.8	0.52
SGP	216	134.5	0.52	225	140.6	0.52
SGP-Ritz	170	120.4	0.52	153	108.9	0.52
512 × 512 × 64						
RL	1568	2158.9	0.49	2500*	3400.0*	0.56*
SGP	351	839.5	0.49	526	1248.2	0.56
SGP-Ritz	219	590.0	0.49	416	1130.7	0.56

sional deconvolution problems faceable almost in real-time. We first provide the technical details of the GPU testing platform and then we report some numerical results to assess the improvements achievable thanks to the GPU implementation of an SGP-based deconvolution algorithm.

4.3.1. GPU testing platform

Nowaday GPUs are very common devices, generally targeted to graphics applications. Their computing-power range is very wide, but we limit our interest to top-powerful models.

As it is well known, modern GPUs are multicore architecture designed to perform a large number of integer/floating-point operations at the same time, follwing a Single-Instrucion Multiple-Threading (SIMT) parallel paradigm. This is an evolution of the classical Single-Instruction Multiple-Data (SIMD) paradigm, which adds to the latter the ability of running multiple processes at the same time on each computing core.

Last-generation Nvidia Fermi GPUs support double precision arithmetic and exploit the powerful 16x PCI-Express (PCIe) connection to the hosting board, which grants up to 16 GB/sec transfer rate.

A Fermi GPU [32] consists of up to 512 CUDA cores. These 512 cores are split across 16 Streaming Multiprocessors (SM), each SM consisting of

32 CUDA cores. The GPU has six 64-bit memory partitions supporting up to 6GB of GDDR5 DRAM memory. Each CUDA core consists of an integer arithmetic logic unit (ALU) and a floating point unit (FPU). In order to support hardware double precision, two cores are jointly used. A Multiprocessor is then equipped with 16 load/store units, thus providing resources for evaluating sixteen source and destination addresses per clock. Moreover, hardware resources are reserved for multiply-add (MAD) instruction, trigonometric function, reciprocal and square root.

Functions that will run on GPU, are both written and launched using CUDA paradigm: for clarity, such kind of functions will be denoted as kernels in the following lines. Each kernel function is executed in a grid of threads. This grid is divided into blocks also known as thread blocks and each block is further divided into threads. This two levels of parallelization are the core approach of both CUDA and any other language aiming at exploiting highly data parallel approach, see for example OpenCL. Programmers can focus on writing kernels that will be run by a single thread, while the hardware scheduler will dispatch millions of them. On the other side, little support is provided for task based parallelism.

Each kernel function, written in order to be run on the GPU, can have access to the system's main global memory where, usually, the problem data are stored. The main memory and the cores are connected by a high-speed 144GB/sec bus. All together, these features provide over 2.5Tflops of system's peak performance for double precision arithmetic.

Given that the GPU-to-GPU communications run through the high-speed channel, while CPU-to-GPU ones go through the PCIe bus, it's easly seen that the latter are almost 10 times slower than the former. This is the reason why to best exploit the GPU computing power one must reduce as most as possible the CPU-to-GPU memory communications. Since 3.0 release, CUDA approach offers a way to overcome the bootleneck that may occur when implementing algorithms that need a continuous data movement between host and device. By reserving a buffer of pinned memory on the CPU, this storage can be targeted for copy to/from GPU in roughly half the time of a standard transfer. On the other side, this approach poses strong requirements on the memory handling (paging, virtual address translation, etc.) by the host OS, and then must be used carefully.

Pinned memory is also a requirement when one has to overlap memory transfer and kernel execution: boards with compute capability above 1.3 can overlap CPU-GPU data exchange and GPU kernel execution. Moreover, since kernel launch is an asynchronous operation, CUDA approach can provide up to 3 different and concurring tasks: device-host transfer, GPU elaboration, and CPU evaluation. In order to exploit these features, all code must be carefully partitioned in independent sections, and data transfer and kernel launch must be issued in different queues, called streams on Nvidia approach. This makes available to the programmer the capacity to overcome the limitations, currently present in GPU architectures, on both task based parallelism and data transfer, but the drawback is the need to manual resolve all synchronization issues that can arise.

The test platform we used for the experiments in this paper is a workstation equipped with 2 Intel Xeon SixCore CPUs at 3.47GHz, 188GB of RAM and 4 GPUs Nvidia Tesla C2070 connected thorugh PCIe channels. The system is operated by a CentOS Linux distribution. Each one of the four GPUs is highly parallel: it embeds 14 streaming multiprocessors, for a total of 448 32-bit computing cores, or 224 double precision cores.

4.3.2. GPU tests

In order to show the remarkable advantages of exploiting the computational power of these architectures in SGP-based deconvolution algorithms, we report the results of several experiments on increasingly large 2D synthetic data sets, generated as described in subsection 4.1. The GPU implementation of the algorithms is obtained by performing on the parallel device the main computational tasks (FFT computations and vector-vector operations) of each iteration. In this way, by running the same number of iterations with the serial and the parallel versions of the deconvolution algorithms, essentially the same reconstruction accuracy is obtained [12, 33, 34]. For each image size, the results reported in Table 4 and 5 are obtained as average values of ten test problems corresponding to different noise realizations. The column denoted by $\tilde{k}$ (std) shows the average number of iterations required by the reconstruction methods to obtain the minimum of the Kullback-Leibler divergence of $\boldsymbol{x}^{(k)}$ from the known image $\tilde{\boldsymbol{x}}$ that is to say

$$\tilde{k} = \underset{k \geq 0}{\operatorname{argmin}}\, D(k) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \tilde{x}_i \ln \left(\frac{\tilde{x}_i}{x_i^{(k)}} \right) + x_i^{(k)} - \tilde{x}_i \right\} ;$$

the standard deviation is reported in brackets. The average values of the minimum Kullback-Leibler divergence and the corresponding standard deviation are shown in the column $D(\tilde{k})$ (std).

The standard SGP version, with step-length selection based on the adaptive alternation of the BB rules, and its GPU implementation presented in [12] are used for these experiments. It is easily seen from Table 4 that for SGP the speed-up (*Sp.* column) in using the GPU version against the CPU version ranges between 3.5 and 22 times. For comparison purpose only, we report in Table 5 the results for the two versions of the RL method. In this case, due to the easier iteration exploited by RL, the time gain ranges between about 7.2 and 16.5; however, despite this fact, the absolute computational time remains considerably larger than that of the SGP method. In particular, the more the image size increases, the more the GPU version of SGP seems preferable, by making reconstructed images available in few seconds also in case of large deconvolution problems. In [12], a 3D deconvolution problem on a true test case with an acquired volume consisting of $1024 \times 1024 \times 33$ voxels is faced with the GPU-based implementation of SGP, obtaining a sensible reconstruction after just 20 iterations, running for about 35 seconds on the architecture described above. Such a result can be considered very promising, given that the collection time for this data set amounts to 180 seconds. Thus, the further convergence rate improvements arising from equipping SGP with the new step-length rule introduced in this work represent a worthwhile contribution toward an effective real-time deconvolution in microcopy. In fact, the convergence rate improvements shown in section 4.2 are obtained by means of the Ritz-like values computation with a small parameter m ($m = 3$), which yields a number of operations (in particular, vector-vector operations) not significantly larger than the number of operations required in the case of the BB-based step-length rule. This is confirmed by the similar average cost per iteration provided by SGP and SGP-Ritz (see Tables 1 and 2). The similar complexity of the two rules allows one to expect that the better convergence rate ensured by the new step-length selection can be fully exploited also on the GPU architecture, for achieving a remarkable gain in the overall reconstruction time.

5. Conclusions and future work

In this paper we introduced a new version of the SGP method equipped with a steplength selection rule based on special Ritz-like values. This rule, recently suggested by R. Fletcher in the context of unconstrained optimization, is here applied in the context of a scaled gradient projection method for (simply) constrained optimization. To this end, we propose a generalization

Table 4: SGP reconstruction time (in seconds) and relative speed-up of the GPU implementation compared with the CPU version.

size	$\tilde{k}$ [av (std)]	$D(\tilde{k})$ [av (std)]	CPU	GPU	Sp.
128×128	98.6 (19.6)	234.4 (27.0)	0.38	0.11	3.5
256×256	87.6 (18.3)	888.2 (128.9)	0.88	0.14	6.3
512×512	79.0 (17.2)	3102.9 (114.7)	2.71	0.27	10.0
1024×1024	57.6 (14.1)	12111.7 (429.8)	9.11	0.59	15.4
2048×2048	63.5 (12.9)	43391.7 (1928.6)	52.58	2.42	21.7
4096×4096	54.9 (6.6)	179775.4 (4116.9)	191.68	8.71	22.0

Table 5: RL reconstruction time (in seconds) and relative speed-up of the GPU implementation compared with the CPU version.

size	$\tilde{k}$ [av (std)]	$D(\tilde{k})$ [av (std)]	CPU	GPU	Sp.
128×128	686.7 (129.0)	254.7 (21.1)	1.01	0.14	7.2
256×256	562.7 (103.2)	1015.9 (137.0)	2.44	0.21	11.6
512×512	521.9 (161.9)	3579.5 (111.0)	8.32	0.69	12.1
1024×1024	506.1 (121.8)	13545.2 (337.8)	41.96	2.40	17.5
2048×2048	458.3 (368.6)	48441.6 (2064.2)	221.45	9.07	24.4
4096×4096	487.9 (189.9)	201250.4 (5267.9)	686.36	41.62	16.5

of the step-length rule able to take into account the presence of scaled gradient directions and nonnegativity constraints. The new rule is able to exploits a cleaver limited-memory implementation, that allows to lower both the computational time and the storage requirements. To asses the advantages of the proposed scheme, we tested it on deconvolution problems coming from 2D and 3D microscopy imaging synthetic data. These experiments show that, compared to the classical BB-based SGP version as well as to the widely used RL method, the new SGP scheme provides meaningful benefits in terms of both number of iterations and computational time, while still getting the same (and sometimes a better) reconstruction error.

Furthermore, we additionally describe how, through a state of the art implementation on today's widely available powerful GPU multicore architectures, the considered schemes allow to speed-up the deconvolution task for large-scale real-world 3D volumes up to very few tens of seconds, which makes the goal of real-time microscopy imaging reconstruction essentially reached.

However, further improvements are still possible from both the theoretical and practical point of view. The strategy exploited in this work to adapt the step-length rule to the presence of nonnegativity constraints is efficient but other ideas can be exploited; in our opinion this topic merits further theoretical investigations and can be worthwhile also in more general constrained optimization framework [35]. From the implementation point of view, the scheme's flexibility and the very appealing features of the new limited-memory rule leave space for additional ways to exploit the power of modern parallel architectures in terms of fast channels communications and cores' multithreading capabilities. We are already working at these aspects, that will be the subject of future works.

References

[1] W. H. Richardson, Bayesian–based iterative method of image restoration, J. Opt. Soc. Amer. A 62 (1) (1972) 55–59.

[2] L. B. Lucy, An iterative technique for the rectification of observed distributions, Astronom. J. 79 (1974) 745–754.

[3] R. Fletcher, A limited memory steepest descent method, Mathematical Programming 135 (1-2) (2012) 413–436.

[4] J. Barzilai, J. Borwein, Two point step size gradient methods, IMA J. Numer. Anal. 8 (1988) 141–148.

[5] S. Bonettini, R. Zanella, L. Zanni, A scaled gradient projection method for constrained image deblurring, Inverse Problems 25.

[6] F. Benvenuto, R. Zanella, L. Zanni, M. Bertero, Nonnegative least-squares image deblurring: improved gradient projection approaches, Inverse Prob. 26 (2) (2010) 025004 (18pp).

[7] W. W. Hager, B. A. Mair, H. Zhang, An affine-scaling interior-point cbb method for box-constrained optimization, Mathematical Programming 119 (2009) 1–32.

[8] S. Setzer, G. Steidl, J. Morgenthaler, A cyclic projected gradient method, Comput. Optim. Appl. 54 (2013) 417–440.

[9] Y. Wang, S. Ma, Projected Barzilai-Borwein method for large-scale nonnegative image restoration, Inverse Problems in Science and Engineering 15 (2007) 559–583.

[10] G. Yu, L. Qi, Y. Dai, On nonmonotone Chambolle gradient projection algorithms for total variation image restoration, Comput. Optim. Appl. 47 (2010) 377–400.

[11] M. Zhu, S. J. Wright, T. F. Chan, Duality-based algorithms for total-variation-regularized image restoration, J. Math. Imaging Vis. 35 (2009) 143–154.

[12] R. Zanella, G. Zanghirati, R. Cavicchioli, L. Zanni, P. Boccacci, M. Bertero, G. Vicidomini, Towards real-time image deconvolution: application to confocal and sted microscopy, Scientific Reports 3.

[13] B. Zhou, L. Gao, Y. H. Dai, Gradient methods with adaptive step-sizes, Computational Optimization and Applications 35 (2006) 69–86.

[14] G. Frassoldati, L. Zanni, G. Zanghirati, New adaptive stepsize selections in gradient methods, J. Industrial and Management Optim. 4 (2008) 299–312.

[15] H. H. Barrett, K. J. Meyers, Foundations of Image Science, Wiley and Sons, 2003.

[16] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Bristol: IoP Publishing, 1998.

[17] P. J. Davis, Circulant Matrices, John Wiley & Sons, Inc., New York, 1979.

[18] I. Csiszár, Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems, Ann. Stat. 19 (1991) 2032–2066.

[19] M. Bertero, P. Boccacci, G. Desiderà, G. Vicidomini, Image deblurring with poisson data: from cells to galaxies, Inverse Problems 25.

[20] K. Lange, R. Carson, EM reconstruction algorithms for emission and transmission tomography, J. Comp. Assisted Tomography 8 (1984) 306–316.

[21] Y. Vardi, L. A. Shepp, L. Kaufman, A statistical model for positron emission tomography, J. Am. Stat. Assoc. 80 (1985) 8–37.

[22] H. N. Mülthei, B. Schorr, On properties of the iterative maximum likelihood reconstruction method, Math. Meth. Appl. Sci. 11 (1989) 331–342.

[23] A. N. Iusem, Convergence analysis for a multiplicatively relaxed EM algorithm, Math. Meth. Appl. Sci. 14 (1991) 573–593.

[24] A. N. Iusem, A short convergence proof of the EM algorithm for a specific Poisson model, REBRAPE 6 (1992) 57–67.

[25] P. Tseng, An analysis of the EM algorithm and entropy–like proximal point methods, Math. Oper. Res. 29 (2004) 27–44.

[26] E. Resmerita, H. Engl, A. Iusem, The expectation-maximization algorithm for ill-posed integral equations: a convergence analysis, Inverse Problems 23 (6) (2007) 2575–2588.

[27] D. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, 1999.

[28] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line-search technique for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.

[29] G. Golub, C. Van Loan, Matrix Computations. 3rd edn., The Johns Hopkins Press, Baltimore, 1996.

[30] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Stand. 45 (1950) 255–282.

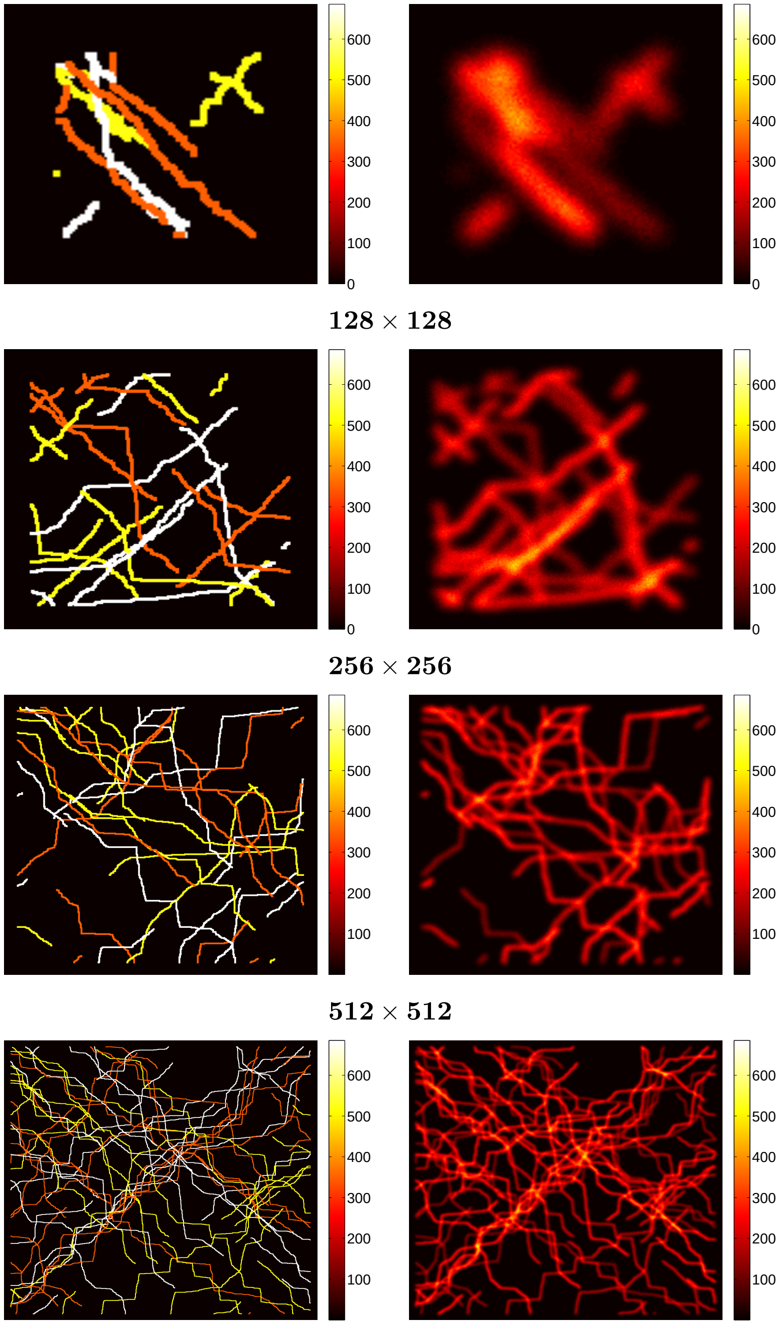
[31] B. Zhang, J. Zerubia, J.-C. Olivo-Marin, Gaussian approximations of fluorescence microscope point-spread function models, Appl. Opt. 46 (10) (2007) 1819–1829. doi:10.1364/AO.46.001819.

[32] NVIDIA Corp., NVIDIA's next generation CUDA compute architecture: Fermi, Whitepaper.
URL \url{http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf}

[33] V. Ruggiero, T. Serafini, R. Zanella, L. Zanni, Iterative regularization algorithms for constrained image deblurring on graphics processors, J. Glob. Optim. 48 (2010) 145–157.

[34] M. Prato, R. Cavicchioli, L. Zanni, P. Boccacci, M. Bertero, Efficient deconvolution methods for astronomical imaging: algorithms and IDL-GPU codes, Astronomy & Astrophysics 539 (2012) A133.

[35] R. Fletcher, A sequential linear constraint programming algorithm for nlp, SIAM Journal on Optimization 22(3) (2012) 772–794.



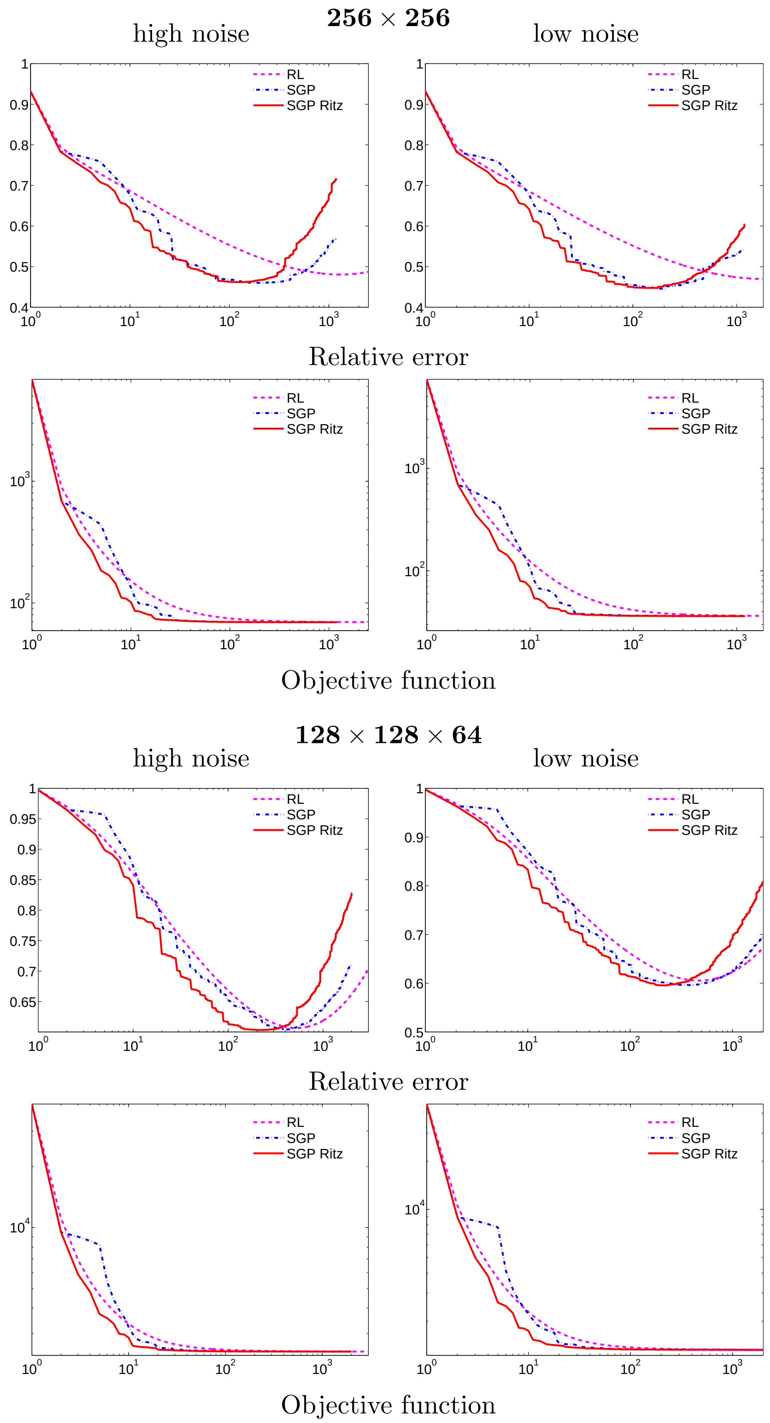
128 × 128

256 × 256

512 × 512

1024 × 1024

Figure 1: Left panel: original images. Right panel: blurred and noisy data (with high noise level).


256 × 256

high noise

low noise

Relative error

Objective function

128 × 128 × 64

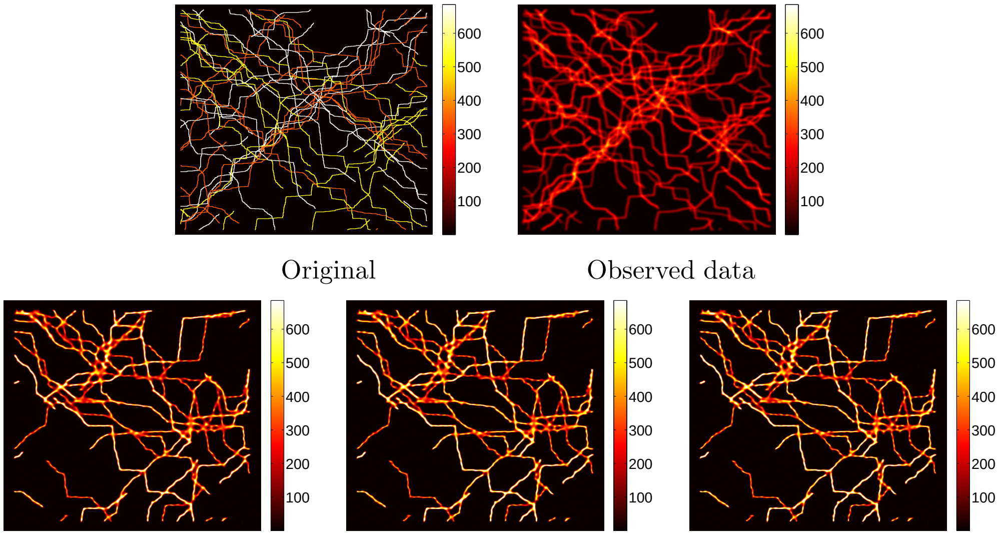
high noise

low noise

Relative error

Objective function

Figure 2: Relative error and decrease of the objective function with respect to the number of iterations.



Original

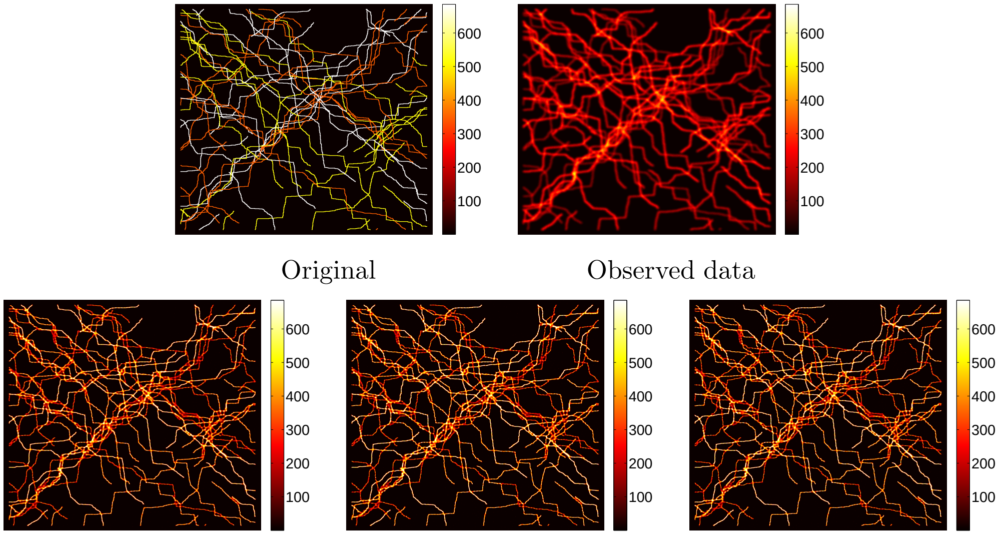
Observed data

RL

SGP

SGP-Ritz

Figure 3: Reconstructions for the high noise level 512 × 512 test problem.



Original

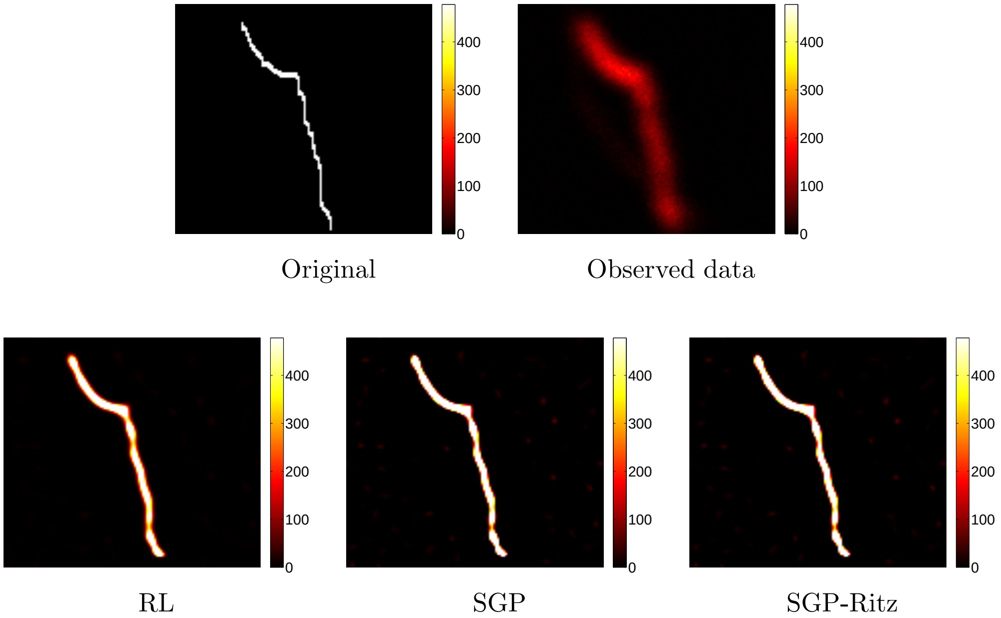
Observed data

RL

SGP

SGP-Ritz

Figure 4: Reconstructions for the low noise level 1024 × 1024 test problem.



Original

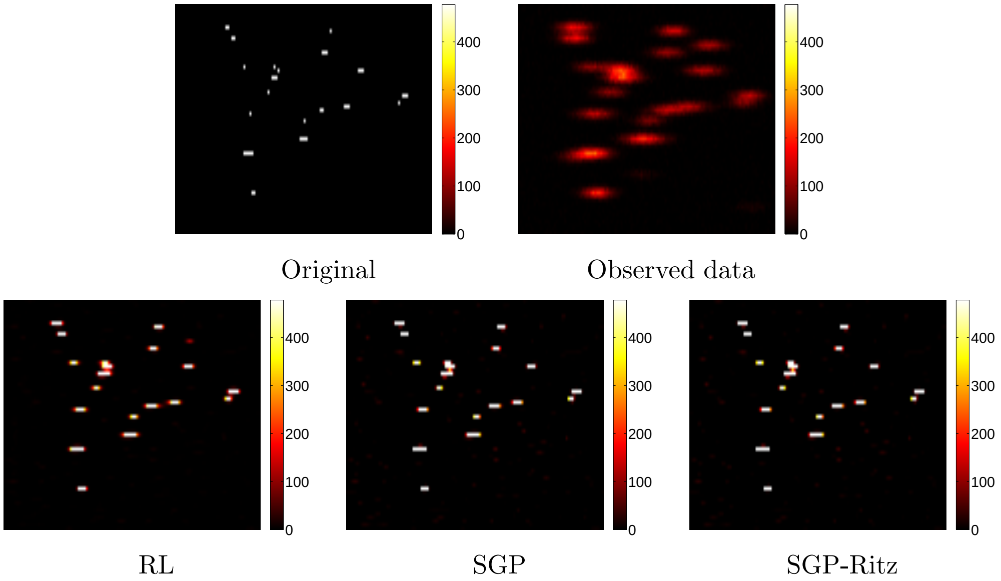
Observed data

RL

SGP

SGP-Ritz

Figure 5: Transverse section reconstruction for the 128 × 128 × 64 image with low noise.



Original

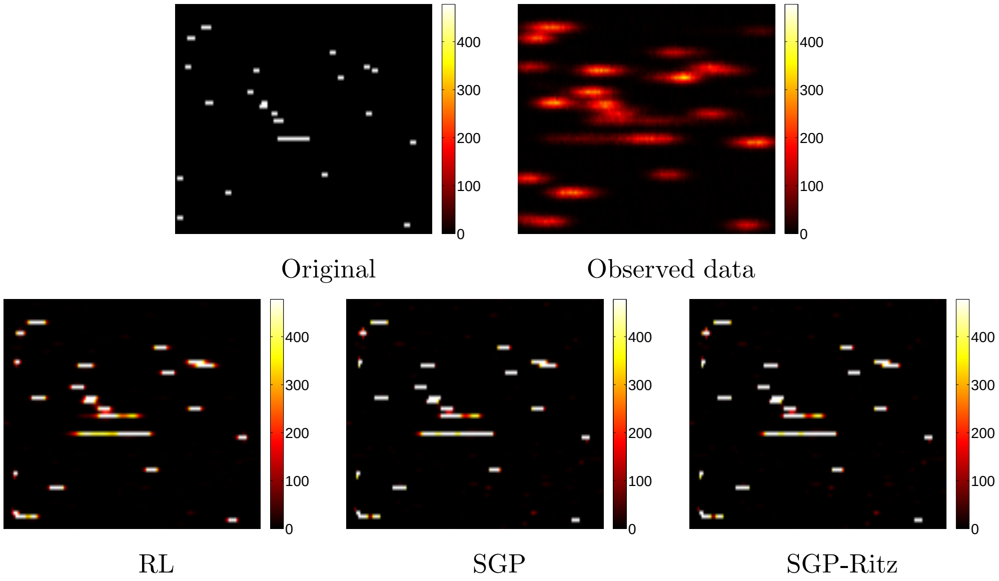
Observed data

RL

SGP

SGP-Ritz

Figure 6: Sagittal section reconstruction for the 128 × 128 × 64 image with low noise.



Original

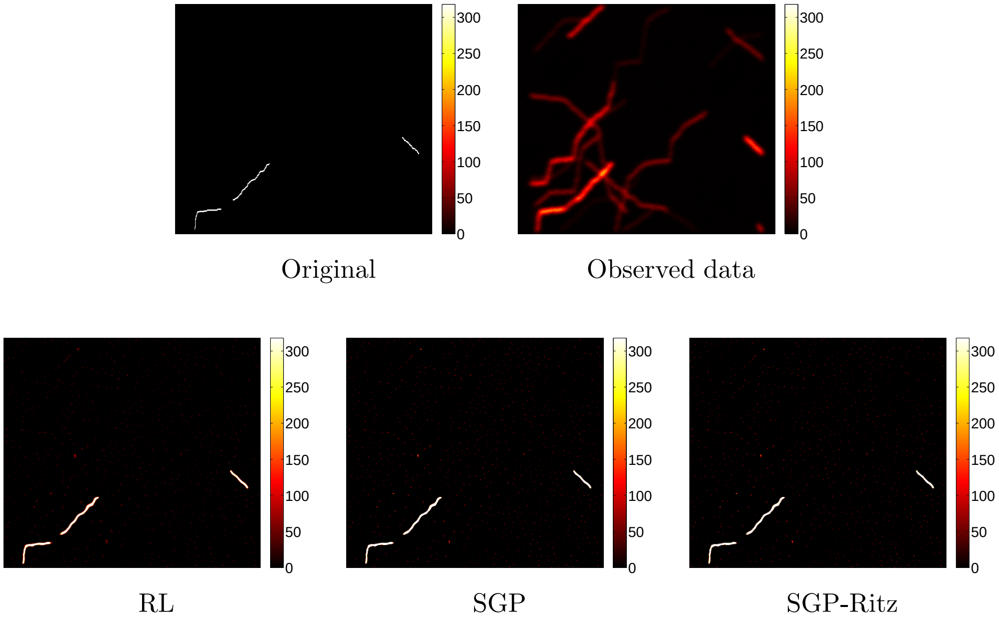
Observed data

RL

SGP

SGP-Ritz

Figure 7: Coronal section reconstruction for the 128 × 128 × 64 image with low noise.



Original

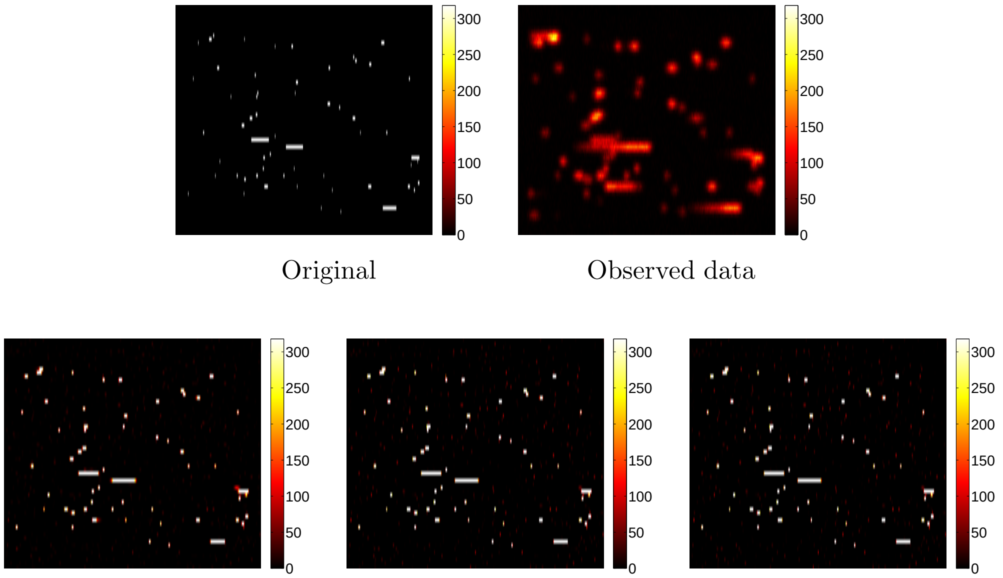
Observed data

RL

SGP

SGP-Ritz

Figure 8: Transversal section reconstruction for the 512 × 512 × 64 image with high noise.



Original

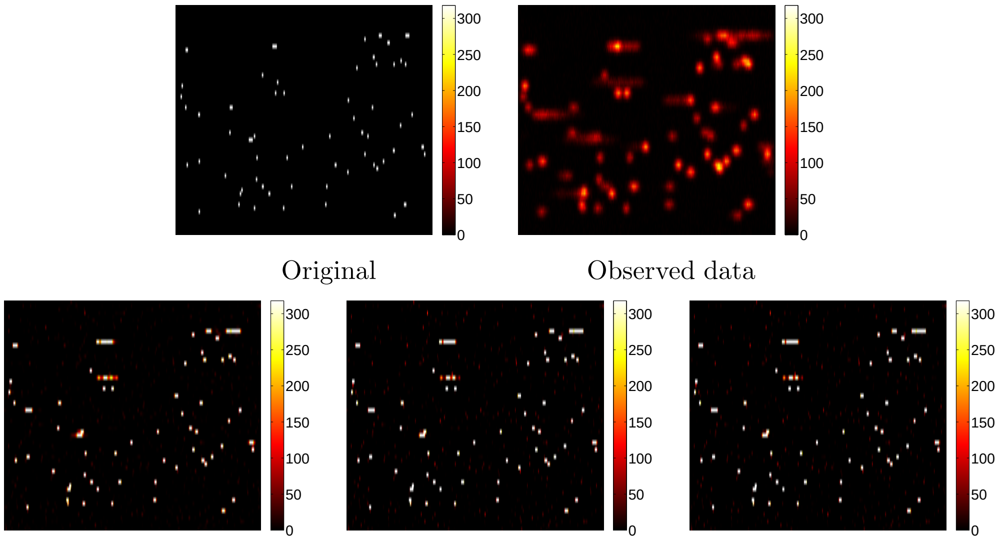
Observed data

RL

SGP

SGP-Ritz

Figure 9: Sagittal section reconstruction for the 512 × 512 × 64 image with high noise.



Original

Observed data

RL

SGP

SGP-Ritz

Figure 10: Coronal section reconstruction for the 512 × 512 × 64 image with high noise.