

A block parallel majorize-minimize memory gradient algorithm

Sara Cadoni, Emilie Chouzenoux, Jean-Christophe Pesquet, Caroline Chaux

► **To cite this version:**

Sara Cadoni, Emilie Chouzenoux, Jean-Christophe Pesquet, Caroline Chaux. A block parallel majorize-minimize memory gradient algorithm. IEEE International Conference on Image Processing (ICIP 2016), Sep 2016, Phoenix, AZ, United States. 2016 IEEE International Conference on Image Processing (ICIP), pp.3194 - 3198, 2016, <10.1109/ICIP.2016.7532949>. <hal-01418417>

HAL Id: hal-01418417

<https://hal.archives-ouvertes.fr/hal-01418417>

Submitted on 16 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

A BLOCK PARALLEL MAJORIZE-MINIMIZE MEMORY GRADIENT ALGORITHM

Sara Cadoni¹, Emilie Chouzenoux¹, Jean-Christophe Pesquet¹, and Caroline Chaux² *

¹ Université Paris-Est Marne-la-Vallée,
LIGM and UMR-CNRS 8049,
Marne-la-Vallée, France
first.last@univ-paris-est.fr

² Aix Marseille Université, CNRS, Centrale Marseille,
I2M and UMR-CNRS 7373,
Marseille, France
caroline.chaux@univ-amu.fr

ABSTRACT

In the field of 3D image recovery, huge amounts of data need to be processed. Parallel optimization methods are then of main interest since they allow to overcome memory limitation issues, while benefiting from the intrinsic acceleration provided by recent multicore computing architectures. In this context, we propose a Block Parallel Majorize-Minimize Memory Gradient (BP3MG) algorithm for solving large scale optimization problems. This algorithm combines a block coordinate strategy with an efficient parallel update. The proposed method is applied to a 3D microscopy image restoration problem involving a depth-variant blur, where it is shown to lead to significant computational time savings with respect to a sequential approach.

Index Terms— Majorization-Minimization ; Block-alternating optimization ; Parallel algorithm ; Memory gradient technique ; 3D deconvolution ; Depth-varying blur.

1. INTRODUCTION

In many inverse problems encountered in image processing, one has to generate an image estimate $\hat{\mathbf{x}} \in \mathbb{R}^N$ by minimizing an appropriate cost function F , which has the following composite form:

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad F(\mathbf{x}) = \sum_{s=1}^S f_s(\mathbf{L}_s \mathbf{x}) \quad (1)$$

where, for every $s \in \{1, \dots, S\}$, $\mathbf{L}_s \in \mathbb{R}^{P_s \times N}$, $P_s \in \mathbb{N}^*$, and f_s is a function from \mathbb{R}^{P_s} to \mathbb{R} . In the context of maximum a posteriori estimation, \mathbf{L}_1 is a degradation operator (for instance, a convolution operator) related to the acquisition process and f_1 measures the fidelity between the data and the observation model in accordance with the noise statistics. The remaining terms $(f_s)_{2 \leq s \leq S}$ act as regularization functions incorporating *a priori* information on the sought solution, or more generally on a linear transform of it, thanks to matrices $(\mathbf{L}_s)_{2 \leq s \leq S}$. Iterative optimization methods must generally be

employed to find a minimizer of (1). In the case of large scale image recovery problems, a major challenge is to design an optimization algorithm able to deliver reliable numerical solutions in a reasonable time.

When all the involved functions $(f_s)_{1 \leq s \leq S}$ are differentiable on \mathbb{R}^N (but non necessarily convex), state-of-the-art smooth optimization methods are nonlinear conjugate gradient (NLCG) and low memory BFGS (L-BFGS) algorithms [1, 2, 3]. The Majorize-Minimize Memory Gradient (3MG) algorithm [4, 5] can be viewed as a special instance of NLCG with optimal stepsize and optimal conjugacy parameter and, when more than one previous direction is employed in the subspace, as a particular L-BFGS algorithm with optimal combination of memory directions. The main advantage of 3MG algorithm is its high efficiency since it relies on a Majorize-Minimize (MM) approach, combined with a subspace acceleration technique. At each iteration, a quadratic majorizing approximation of (1) at the current iterate is constructed, and the next iterate results from its minimization within a two-dimensional subspace spanned by the current gradient and the previous direction. In addition, the 3MG algorithm enjoys nice convergence properties which have been investigated in [4] in the convex case, and in [6] in the non-convex case. Comparisons with graph-cut based discrete optimization methods, and primal-dual proximal algorithms on a number of image restoration problems have also shown the good performance of 3MG in terms of practical convergence speed [6, 7, 8, 9]. However, when the size of the problem becomes increasingly large, as it may happen in 3D image processing or video processing, running this kind of algorithm becomes difficult, due to memory limitation issues.

An efficient way to overcome difficulties related to memory requirements in optimization algorithm consists in adopting a parallel block alternating minimization approach [10, 11, 12]. The target vector is split into non-overlapping blocks of reduced dimension and, at each iteration of the algorithm, few blocks are selected, and updated in a parallel manner according to a given optimization scheme. This strategy allows to control the required memory, since the number of blocks can be set by the user to the number of available processors.

Several works in the context of image restoration, have

*This work was supported by the CNRS Imag'In project under grant 2015 OPTIMISME.

shown that MM algorithms benefit from a high potential of parallelization [13, 14]. These studies are based on the use of majorant functions having a particular block separable structure that leads inherently to parallel update schemes which may be of great interest when the optimization is performed on multicore systems or GPGPU boards. The usage of block alternating techniques in MM strategies has also been investigated (see, for instance, [15, 16]). However, up to our knowledge, the resulting methods do not present any parallel structure, as only one block must be updated per iteration, so that the gain in terms of memory usage is usually obtained at the expense of a substantial decrease of the convergence rate [13].

In this work we propose a new Block Parallel 3MG (BP3MG) algorithm for smooth, possibly nonconvex, optimization problems involving massive datasets. The proposed method has two key benefits: it takes advantage of recent technological advances in parallel computing on multicore architectures and it allows the computation load to be tailored to the number of available processors. We show that these new features result in a highly accelerated method for solving a 3D deblurring problem with depth-variant blur.

The paper is organized as follows: in Section 2 we introduce the proposed BP3MG algorithm. In particular we describe the block subspace and the block separable majorant function used for its derivation. Section 3 describes a parallel implementation of this method for the restoration of 3D microscopy images and shows its very good performance. Some conclusions are drawn in Section 4.

2. PROPOSED METHOD

2.1. Notation

In our block-coordinate approach, the vector of unknowns $\mathbf{x} = (x_n)_{1 \leq n \leq N} \in \mathbb{R}^N$ is split into J vectors $\mathbf{x}^{(j)}$ of reduced dimension $N_j \neq 0$ with $j \in \{1, \dots, J\}$ ($N_1 + \dots + N_J = N$). We will also need to define subsets $\mathcal{S} \subset \{1, \dots, J\}$ gathering such block indices as illustrated in Fig. 1. The vector of variables associated with a set \mathcal{S} is then denoted by $\mathbf{x}^{(\mathcal{S})} = (\mathbf{x}^{(j)})_{j \in \mathcal{S}}$ and its components are $(x_p)_{p \in \mathcal{S}}$ where $\mathbb{S} \subset \{1, \dots, N\}$ with cardinality $|\mathbb{S}| = \sum_{j \in \mathcal{S}} N_j$.

For every $s \in \{1, \dots, S\}$, the matrix \mathbf{L}_s in (1) can be decomposed as $[\mathbf{L}_s^{(1)} | \dots | \mathbf{L}_s^{(J)}]$, where, for every $j \in \{1, \dots, J\}$, $\mathbf{L}_s^{(j)} \in \mathbb{R}^{P_s \times N_j}$. Finally, we denote by $\mathbf{L}_s^{(\mathcal{S})} \in \mathbb{R}^{P_s \times |\mathcal{S}|}$, the columnwise concatenation of matrices $(\mathbf{L}_s^{(j)})_{j \in \mathcal{S}}$. A similar notation is used for the (rowwise) block decomposition of the gradient $\nabla F(\mathbf{x})$ at some given vector $\mathbf{x} \in \mathbb{R}^N$.

2.2. Block separable quadratic majorant function

The majorize-minimize approaches [17, 18, 19] often rely on the use of a surrogate function which constitutes a quadratic *tangent majorant* functions for Criterion (1). Let $\tilde{\mathbf{x}} \in \mathbb{R}^N$

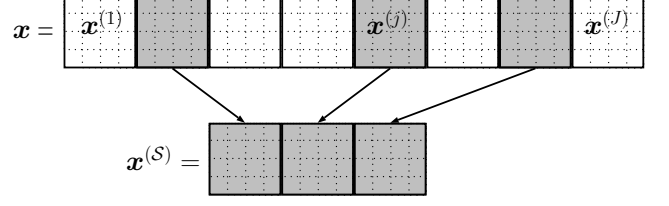


Fig. 1. Example of block subset, with $\text{Card}(\mathcal{S}) = 3$. Each block has here 16 components.

and, for every $\mathbf{x} \in \mathbb{R}^N$, let $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^N \times \mathbb{R}^N$ be a symmetric positive matrix. The quadratic function

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad Q(\mathbf{x}, \tilde{\mathbf{x}}) = F(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}})^\top (\mathbf{x} - \tilde{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \tilde{\mathbf{x}})^\top \mathbf{A}(\tilde{\mathbf{x}}) (\mathbf{x} - \tilde{\mathbf{x}}),$$

is said to be a *tangent majorant* for F at $\tilde{\mathbf{x}}$, if

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad Q(\mathbf{x}, \tilde{\mathbf{x}}) \geq F(\mathbf{x}) \quad \text{and} \quad Q(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) = F(\tilde{\mathbf{x}}). \quad (2)$$

In the sequel we will assume the existence of matrices fulfilling (2) of the form

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathbf{A}(\mathbf{x}) = \sum_{s=1}^S \mathbf{L}_s^\top \text{Diag} \{ \omega_s(\mathbf{L}_s \mathbf{x}) \} \mathbf{L}_s, \quad (3)$$

where, for every $s \in \{1, \dots, S\}$, \mathbf{L}_s is a $P_s \times N$ real-valued matrix and ω_s is a function from \mathbb{R}^{P_s} to $]0, +\infty[^{P_s}$ whose expression depends on the properties of the potential f_s [5, Tab. I]. For instance, if f_s is β_s -Lipschitz differentiable with $\beta_s \in]0, +\infty[$, we can choose $\omega_s(\cdot) = \beta_s \mathbf{1}_{P_s}$, where $\mathbf{1}_{P_s}$ is a vector with length P_s whose components are all equal to one.

In order to derive our block-coordinate algorithm, we need to define, for every nonempty set of blocks $\mathcal{S} \in \{1, \dots, J\}$, a quadratic majorant function of the restriction of F to these blocks. According to [20, Rem. 2.4], for every $\tilde{\mathbf{x}} \in \mathbb{R}^N$, a possible choice is

$$(\forall \mathbf{v} \in \mathbb{R}^{|\mathcal{S}|}) \quad Q_{\mathbf{A}^{(\mathcal{S})}}^{(\mathcal{S})}(\mathbf{v}, \tilde{\mathbf{x}}) = F(\tilde{\mathbf{x}}) + \nabla F^{(\mathcal{S})}(\tilde{\mathbf{x}})^\top (\mathbf{v} - \tilde{\mathbf{x}}^{(\mathcal{S})}) + \frac{1}{2} (\mathbf{v} - \tilde{\mathbf{x}}^{(\mathcal{S})})^\top \mathbf{A}^{(\mathcal{S})}(\tilde{\mathbf{x}}) (\mathbf{v} - \tilde{\mathbf{x}}^{(\mathcal{S})}), \quad (4)$$

where

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathbf{A}^{(\mathcal{S})}(\mathbf{x}) = ([\mathbf{A}(\mathbf{x})]_{p,p})_{p \in \mathcal{S}}. \quad (5)$$

However, in practice, the matrices defined in (5) do not necessarily have a block diagonal structure so that $Q_{\mathbf{A}^{(\mathcal{S})}}^{(\mathcal{S})}(\cdot, \tilde{\mathbf{x}})$ is not separable with respect to each block $j \in \mathcal{S}$. Therefore, its minimization cannot be performed efficiently in a parallel manner. The following lemma defining block diagonal matrices majorizing (5) will be the cornerstone of the parallel structure of our algorithm:

Lemma 1. Let $\mathcal{S} \subset \{1, \dots, J\}$. For every $j \in \mathcal{S}$, for every $\mathbf{x} \in \mathbb{R}^N$, let matrix $\mathbf{B}^{(j)}(\mathbf{x}) \in \mathbb{R}^{N_j \times N_j}$ be given by

$$\mathbf{B}^{(j)}(\mathbf{x}) = \sum_{s=1}^S \left((\mathbf{L}_s^{(j)})^\top \text{Diag} \{ \mathbf{b}_s(\mathbf{L}_s \mathbf{x}) \} \mathbf{L}_s^{(j)} \right), \quad (6)$$

where, for every $s \in \{1, \dots, S\}$ and $p \in \{1, \dots, P_s\}$, the p -th component of vector $\mathbf{b}_s(\mathbf{L}_s \mathbf{x})$ is¹

$$[\mathbf{b}_s(\mathbf{L}_s \mathbf{x})]_p = [\boldsymbol{\omega}_s(\mathbf{L}_s \mathbf{x})]_p / [|\mathbf{L}_s^{(S)}| \mathbf{1}_{|\mathcal{S}|}]_p / [|\mathbf{L}_s^{(j)}| \mathbf{1}_{N_j}]_p. \quad (7)$$

Then, for every $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{A}^{(S)}(\mathbf{x}) \preceq \mathbf{B}^{(S)}(\mathbf{x})$, where $\mathbf{B}^{(S)}(\mathbf{x}) = \text{BDiag} \left\{ (\mathbf{B}^{(j)}(\mathbf{x}))_{j \in \mathcal{S}} \right\}$ (i.e. the block diagonal matrix with $|\mathcal{S}|$ blocks equal to $\mathbf{B}^{(j)}(\mathbf{x})$, $j \in \mathcal{S}$).

The proof of Lemma 1 is omitted due to the lack of space. It makes a clever use of Jensen's inequality, following techniques similar to those in [21, Sec. III] (see also [13, Sec. III-A]).

2.3. BP3MG algorithm

We are now ready to present our algorithm. At each iteration $k \in \mathbb{N}$, a subset $\mathcal{S}_k \subset \{1, \dots, J\}$ of block indices with cardinality $C \in \mathbb{N}^*$, is selected. C typically corresponds to the number of available processors. The corresponding blocks are then updated thanks to a local 3MG iteration, corresponding to the minimization of $Q_{\mathbf{B}^{(\mathcal{S}_k)}}^{(\mathcal{S}_k)}(\cdot, \mathbf{x}_k)$ within the subspace spanned by the columns of $\mathbf{D}_k^{(\mathcal{S})} = \text{BDiag} \{ (\mathbf{D}_k^{(j)})_{j \in \mathcal{S}} \}$ where, for every $k \in \mathbb{N}$ and $j \in \mathcal{S}_k$,

$$\mathbf{D}_k^{(j)} = \begin{cases} -\nabla F^{(j)}(\mathbf{x}_k) & \text{if } j \notin \bigcup_{\ell=0}^{k-1} \mathcal{S}_\ell, \\ [-\nabla F^{(j)}(\mathbf{x}_k)] \mathbf{x}_k^{(j)} - \mathbf{x}_{k-1}^{(j)} & \text{otherwise.} \end{cases} \quad (8)$$

The above condition on j simply allows us to distinguish blocks which have never been updated, from the blocks which have been updated at least once. Thanks to the block-separable structure of the majorant functions, the C selected blocks with indices $j \in \mathcal{S}_k$ can be updated in a parallel manner, leading to the following new algorithm:

Initialize $\mathbf{x}_0 \in \mathbb{R}^N$.

For $k = 0, 1, 2, \dots$

$$\left[\begin{array}{l} \mathbf{Select} \quad \mathcal{S}_k \subset \{1, \dots, J\} \quad \text{s.t.} \quad |\mathcal{S}_k| = C \\ \mathbf{Parfor} \quad j \in \mathcal{S}_k \\ \quad \text{Compute} \quad \nabla F^{(j)}(\mathbf{x}_k) \\ \quad \text{Compute} \quad \mathbf{B}_k^{(j)}(\mathbf{x}_k) \quad \text{using (6)} \\ \quad \text{Construct} \quad \mathbf{D}_k^{(j)} \quad \text{using (8)} \\ \quad \mathbf{u}_k^{(j)} = - \left((\mathbf{D}_k^{(j)})^\top \mathbf{B}_k^{(j)}(\mathbf{x}_k) \mathbf{D}_k^{(j)} \right)^\dagger (\mathbf{D}_k^{(j)})^\top \nabla F^{(j)}(\mathbf{x}_k) \\ \quad \mathbf{x}_{k+1}^{(j)} = \mathbf{x}_k^{(j)} + \mathbf{D}_k^{(j)} \mathbf{u}_k^{(j)} \\ \mathbf{Set}, \text{ for every } j \in \{1, \dots, J\} \setminus \mathcal{S}_k, \quad \mathbf{x}_{k+1}^{(j)} = \mathbf{x}_k^{(j)}. \\ \mathbf{Share} \quad (\mathbf{x}_{k+1}^{(j)})_{j \in \mathcal{S}_k} \quad \text{between all cores.} \end{array} \right. \quad (9)$$

¹The notation $|M|$ is used to denote the matrix whose elements are equal to the absolute value of those of M .

Hereabove, $(\cdot)^\dagger$ denotes the pseudo inverse operation. The selection rule for the blocks plays an important role in the convergence analysis. Actually, it suffices to assume a quasi-cyclic rule, where the blocks are updated in an arbitrary order, as soon as each one is updated at least once per a finite number $K \in \mathbb{N}^*$ of iterations. Under this assumption, the monotonic convergence of the criterion sequence $(F(\mathbf{x}_k))_{k \in \mathbb{N}}$ to a (locally) optimal value can be established, using the same theoretical tools as in [20].

It is important to emphasize, that, because of Expressions (1) and (6), in practice, it is not necessary to send the full vector \mathbf{x}_{k+1} to all the cores, at iteration k . Actually, let $j \in \{1, \dots, J\}$ and let \mathbb{J}_j be the index set of the components of vector \mathbf{x}_{k+1} belonging to the j -th block. Only the components of \mathbf{x}_{k+1} with indices in

$$\mathcal{N}_j = \bigcup_{s=1}^S \{n \in \{1, \dots, N\} | (\exists p \in \mathcal{P}_{s,j}) [\mathbf{L}_s]_{p,n} \neq 0\},$$

where $\mathcal{P}_{s,j} = \{p \in \{1, \dots, P_s\} | (\exists j \in \mathbb{J}_j) [\mathbf{L}_s]_{p,j} \neq 0\}$, are required to compute the gradient and the majorant matrix that are useful for the update of the j -th block. The cardinality of \mathcal{N}_j is usually very small with respect to N . For instance, if $S = 1$ and \mathbf{L}_1 is a discrete gradient operator with one pixel neighborhood, then $|\mathcal{N}_j| = 3$.

3. APPLICATION TO 3D IMAGE DECONVOLUTION

3.1. Observation model

We focus on the following linear degradation model:

$$\mathbf{y} = \mathbf{H} \bar{\mathbf{x}} + \mathbf{b}, \quad (10)$$

where $\bar{\mathbf{x}} \in \mathbb{R}^N$ represents the 3D original (unknown) image of size $N = N_X \times N_Y \times N_Z$, $\mathbf{y} \in \mathbb{R}^N$ is the observed image, \mathbf{H} is a linear operator modeling a 3D convolution with depth variant blur, and \mathbf{b} is a realization of an additive random noise. The goal is to solve the associated inverse problem, i.e. to find an estimate of $\bar{\mathbf{x}}$ from \mathbf{y} .

3.2. Objective function

A hybrid penalized least squares criterion involving a smooth 3D regularization term is used for the objective function F [9]. Then, F takes the form (1) with

- $P_1 = N$, $\mathbf{L}_1 = \mathbf{H}$, $f_1 = \frac{1}{2} \|\cdot - \mathbf{y}\|^2$,
- $P_2 = N$, $\mathbf{L}_2 = \mathbf{I}_N$, $f_2 = \eta d_{[x_{\min}, x_{\max}]}^2$,
- $P_3 = 2N$, $\mathbf{L}_3 = [(\mathbf{V}^X)^\top (\mathbf{V}^Y)^\top]^\top$,
 $f_3(\mathbf{L}_3 \mathbf{x}) = \lambda \sum_{n=1}^N \sqrt{([\mathbf{V}^X \mathbf{x}]_n)^2 + ([\mathbf{V}^Y \mathbf{x}]_n)^2 + \delta^2}$,
- $P_4 = N$, $\mathbf{L}_4 = \mathbf{V}^Z$, $f_4 = \kappa \|\cdot\|^2$,

where $(\eta, \lambda, \delta, \kappa) \in (0, +\infty)^4$, $\mathbf{V}^X \in \mathbb{R}^{N \times N}$, $\mathbf{V}^Y \in \mathbb{R}^{N \times N}$, $\mathbf{V}^Z \in \mathbb{R}^{N \times N}$ are discrete gradient operators along X, Y, and Z directions, d_E is the distance to a set E , and $(x_{\min}, x_{\max}) \in \mathbb{R}^2$ are some minimal and maximal bounds on the sought intensity values. The same matrices as in [9] are used for defining (3).

3.3. Implementation

The BP3MG algorithm derived in Section 2 has been implemented in MATLAB[®] (2015b) using the Message Passing Interface command *SPMD* of the Parallel Computing Toolbox[™]. More precisely, the blocks correspond to the $J = N_Z$ 2D slices of the 3D volume. For a given number of active cores \bar{C} , the first core is used as a master process while the $C = \bar{C} - 1$ remaining cores are used as slave processes. The master process carries out the main loop of Algorithm (9). At each iteration $k \in \mathbb{N}$, it selects C block indices, stored in \mathcal{S}_k , and sends to each slave the required data allowing it to update the j -th block with $j \in \mathcal{S}_k$. The slave processes perform their tasks simultaneously to compute the variables $(\mathbf{x}_{k+1}^{(j)})_{j \in \mathcal{S}_k}$, and finally send their results to the master.

3.4. Experimental results

We considered two image restoration scenarii described by Model (10). The 3D microscopical images *FlyBrain*² with size $N = 256 \times 256 \times 48$ and *Tube*³ with size $N = 284 \times 280 \times 48$ are considered for \bar{x} . In both cases, the linear operator \mathbf{H} models a 3D depth-varying Gaussian blur, with kernel size $5 \times 5 \times 11$. For each depth $z \in \{1, \dots, N_Z\}$, the blur kernel is characterized by different variance and rotation parameters $(\sigma_X(z), \sigma_Y(z), \sigma_Z(z), \varphi_X(z), \varphi_Z(z))$, following the model from [22]. In practice, the values of these five parameters are chosen randomly and independently, following a uniform distribution for every z , leading to a blurred signal to noise ratio (BSNR) equal to 14.09 dB (resp. 12.58 dB). A zero-mean white Gaussian noise with standard deviation 0.02 is then added to the blurred volume, leading to a SNR equal to 13.42 dB (resp. 11.53 dB). The regularization parameters $(\lambda, \delta, \kappa, \eta)$, and $(x_{\min}, x_{\max}) = (0, 1)$ are chosen so as to maximize the SNR of the restored volume. Here, we obtain a final SNR equal to 16.98 dB (resp. 14.47 dB). Fig. 2 displays some examples of original, degraded and restored slices, for two slices of *FlyBrain*. In Fig. 3, we plot, for different values of the number of active cores \bar{C} , the ratio between the required computation time for one core versus the actual computational time, for reaching the stopping criterion $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| \leq 10^{-5} \|\mathbf{x}_k\|$. The computations were performed on an Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz with 24 physical cores. The result of a linear fitting illus-

trates the great potential of parallelization of the proposed algorithm.

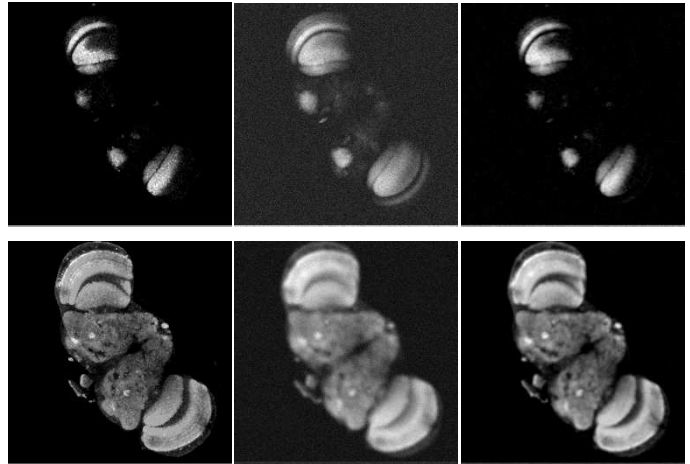


Fig. 2. Original (left), degraded (middle) and restored (right) images corresponding to slices $z = 6$ (top) and $z = 18$ (bottom) of the 3D volume *FlyBrain*.

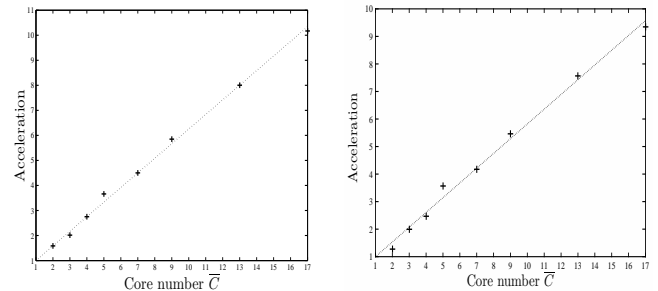


Fig. 3. Ratio between the computation time for one core and the computation time for \bar{C} cores (crosses) with linear fitting (dotted line), for the restoration of *FlyBrain* (left) and *Tube* (right).

4. CONCLUSION

In this paper, we have presented a new Block Parallel Majorize-Minimize Memory Gradient algorithm for handling large-size optimization problems such as those encountered in 3D image restoration with depth-variant blur. In this context, our experimental results illustrate the high efficiency of the proposed method in terms of acceleration for multi-core architectures. Our future work will focus on the implementation in other languages that may lead to a further improved performance.

²<http://imagej.nih.gov/ij/plugins/3d-viewer/>

³<http://adm.irbbarcelona.org/image-j-fiji>

5. REFERENCES

- [1] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, 1989.
- [2] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pac. J. Optim.*, vol. 2, no. 1, pp. 35–58, Jan. 2006.
- [3] W. Chen, W. Zhenghao, and J. Zhou, "Large-scale L-BFGS using MapReduce," in *Proc. Adv. Conf. Neur. Inform. Proc. Syst. (NIPS 2014)*, Montréal, Canada, 8–13 Dec. 2014, pp. 1332–1340.
- [4] E. Chouzenoux, J. Idier, and S. Moussaoui, "A Majorize-Minimize strategy for subspace optimization applied to image restoration," *IEEE Trans. Image Process.*, vol. 20, no. 18, pp. 1517–1528, Jun. 2011.
- [5] E. Chouzenoux and J.-C. Pesquet, "A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation," Tech. Rep., 2015, <http://arxiv.org/abs/1512.08722>.
- [6] E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot, "A majorize-minimize subspace approach for ℓ_2 - ℓ_0 image regularization," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 563–591, 2013.
- [7] E. Chouzenoux, F. Zolyniak, E. Gouillart, and H. Talbot, "A majorize-minimize memory gradient algorithm applied to X-ray tomography," in *Proc. 20th IEEE Int. Conf. Image Process. (ICIP 2013)*, Melbourne, Australia, 15–18 Sep. 2013, pp. 1011–1015.
- [8] A. Florescu, E. Chouzenoux, J.-C. Pesquet, P. Ciuciu, and S. Ciochina, "A majorize-minimize memory gradient method for complex-valued inverse problems," *Sig. Process.*, vol. 103, pp. 285–295, 2014.
- [9] E. Chouzenoux, L. Lamassé, S. Anthoine, C. Chaux, A. Jaouen, I. Vanzetta, and F. Debarbieux, "Approche variationnelle pour la déconvolution rapide de données 3D en microscopie biphotonique," in *Actes du 25e colloque GRETSI*, Lyon, France, 8–11 Sep. 2015.
- [10] A. Repetti, E. Chouzenoux, and J.-C. Pesquet, "A parallel block-coordinate approach for primal-dual splitting with arbitrary random block selection," in *Proc. 23rd Eur. Sig. Process. Conf. (EUSIPCO 2015)*, Nice, France, 1–5 Sep. 2015, pp. 235–239.
- [11] J. Konečný, H. B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," Tech. Rep., 2015, <http://arxiv.org/pdf/1511.03575v1.pdf>.
- [12] A. Onose, R. E. Carrillo, A. Repetti, J. D. McEwen, J.-T. Thiran, J.-C. Pesquet, and Y. Wiaux, "Scalable splitting algorithms for big-data interferometric imaging in the SKA era," Tech. Rep., 2016, <http://arxiv.org/abs/1601.04026>.
- [13] S. Sotthivirat and J. A. Fessler, "Image recovery using partitioned-separable paraboloidal surrogate coordinate ascent algorithms," *IEEE Trans. Signal Process.*, vol. 11, no. 3, pp. 306–317, 2002.
- [14] M. Legendre, S. Moussaoui, E. Chouzenoux, and J. Idier, "Primal-dual interior-point optimization based on majorization-minimization for edge preserving spectral unmixing," in *Proc. 21st IEEE Int. Conf. Image Process. (ICIP 2014)*, Paris, France, 27–30 Oct. 2014, pp. 4161–4165.
- [15] H. Erdogan and J. A. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, no. 9, pp. 801–814, Sept. 1999.
- [16] M. Hong, M. Razaviyayn, Z. Q. Luo, and J. S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.
- [17] M. W. Jacobson and J. A. Fessler, "An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2411–2422, Oct. 2007.
- [18] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Stat.*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
- [19] Z. Zhang, J. T. Kwok, and D.-Y. Yeung, "Surrogate maximization/minimization algorithms and extensions," *Mach. Learn.*, vol. 69, pp. 1–33, 2007.
- [20] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward-backward algorithm," *To appear in J. Global Optim.*, 2015, http://www.optimization-online.org/DB_HTML/2013/12/4178.html.
- [21] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 132–137, 1995.
- [22] O. Wirjadi and T. Breuel, "Approximate separable 3D anisotropic gauss filter," in *Proc. 12nd IEEE Int. Conf. Image Process. (ICIP 2005)*, Genova, Italy, 11–14 Sept 2005, vol. 2, pp. 149–52.