Optimization approaches for image reconstruction on multiprocessor systems

Riccardo Zanella

Laboratorio per le Tecnologie delle Terapie Avanzate (LTTA)
Università degli Studi di Ferrara

Genova, February 2, 2011

Joint work with:

Mario Bertero
Roberto Cavicchioli
Andrea Prearo
Luca Zanni

Silvia Bonettini
Ambra Giovannini
Riccardo Zanella
Gaetano Zanghirati

Problem setting

Constrained optimization problem

$$\min f(\boldsymbol{x})$$
$$\text{sub. to } \boldsymbol{x} \in \Omega$$

Problem setting

Constrained optimization problem

$$\min f(\boldsymbol{x})$$
$$\text{sub. to } \boldsymbol{x} \in \Omega$$

Ω	is a convex and closed set
$f(\boldsymbol{x})$	is countinuously differentiable in $\Omega \subset \mathbb{R}^n$

Why gradient type methods?

Gradient methods are first-order iterative optimization methods.

Why gradient type methods?

Gradient methods are first-order iterative optimization methods.

pros

- Simplicity of implementation
  compute: function, gradient, and $\mathcal{O}(n)$ ops. only, no Hessian
- Low memory requirements
  suitable to face high dimensional problems ($n \gg$)
- Ability to provide solutions with medium-to-high accuracy
- Semiconvergence
  allow the iteration counter to act as the reg. par.
- Very effective when the feasible set Ω is "simple"

Why gradient type methods?

Gradient methods are first-order iterative optimization methods.

pros

- Simplicity of implementation

  compute: function, gradient, and $\mathcal{O}(n)$ ops. only, no Hessian
- Low memory requirements

  suitable to face high dimensional problems ($n \gg$)
- Ability to provide solutions with medium-to-high accuracy
- Semiconvergence

  allow the iteration counter to act as the reg. par.
- Very effective when the feasible set Ω is “simple”

cons

- Low convergence rate

  hundreds or thousands of iterations to full convergence, but acceleration techniques exist

Accelerating gradient methods

The Barzilai-Borwein (BB) step-length selection rules

Consider the general gradient method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)} \qquad k = 0, 1, \dots \ ,$$

with $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Accelerating gradient methods

The Barzilai-Borwein (BB) step-length selection rules

Consider the general gradient method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)} \qquad k = 0, 1, \dots ,$$

with $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Problem:

Can $\alpha_k > 0$ be chosen to improve the convergence rate?

Accelerating gradient methods

The Barzilai-Borwein (BB) step-length selection rules

Consider the general gradient method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)} \qquad k = 0, 1, \ldots ,$$

with $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Solution:

Look at $B(\alpha_k) = (\alpha_k I)^{-1}$ as an easy approximation of $\nabla^2 f(\boldsymbol{x}^{(k)})$

Accelerating gradient methods

The Barzilai-Borwein (BB) step-length selection rules

Consider the general gradient method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)} \qquad k = 0, 1, \dots \ ,$$

with $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Solution:

Look at $B(\alpha_k) = (\alpha_k I)^{-1}$ as an easy approximation of $\nabla^2 f(\boldsymbol{x}^{(k)})$

Determine α_k by forcing a quasi-Newton property on $B(\alpha_k)$:

$$\alpha_k{}^{\mathrm{BB1}} = \arg\min_{\alpha \in \mathbb{R}} \| B(\alpha) \boldsymbol{s}^{(k-1)} - \boldsymbol{z}^{(k-1)} \|$$

or

$$\alpha_k{}^{\mathrm{BB2}} = \arg\min_{\alpha \in \mathbb{R}} \| \boldsymbol{s}^{(k-1)} - B(\alpha)^{-1} \boldsymbol{z}^{(k-1)} \|,$$

where $\boldsymbol{s}^{(k-1)} = \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right)$ and $\boldsymbol{z}^{(k-1)} = \left(\boldsymbol{g}^{(k)} - \boldsymbol{g}^{(k-1)}\right)$.

Accelerating gradient methods

The Barzilai-Borwein (BB) step-length selection rules

Consider the general gradient method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)} \qquad k = 0, 1, \dots ,$$

with $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Solution:

Look at $B(\alpha_k) = (\alpha_k I)^{-1}$ as an easy approximation of $\nabla^2 f(\boldsymbol{x}^{(k)})$

Determine α_k by forcing a quasi-Newton property on $B(\alpha_k)$:

$$\alpha_k^{\text{BB1}} = \frac{\boldsymbol{s}^{(k-1)^T} \boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T} \boldsymbol{z}^{(k-1)}}$$

or

$$\alpha_k^{\text{BB2}} = \frac{\boldsymbol{s}^{(k-1)^T} \boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T} \boldsymbol{z}^{(k-1)}}$$

where $\boldsymbol{s}^{(k-1)} = \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right)$ and $\boldsymbol{z}^{(k-1)} = \left(\boldsymbol{g}^{(k)} - \boldsymbol{g}^{(k-1)}\right)$.

Accelerating gradient methods

The Barzilai-Borwein (BB) step-length selection rules

Consider the general scaled gradient method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k D_k \boldsymbol{g}^{(k)} \qquad k = 0, 1, \dots ,$$

with $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Solution:

Look at $B(\alpha_k) = (\alpha_k D_k)^{-1}$ as an easy approximation of $\nabla^2 f(\boldsymbol{x}^{(k)})$

Determine α_k by forcing a quasi-Newton property on $B(\alpha_k)$:

$$\alpha_k^{\text{BB1}} = \frac{\boldsymbol{s}^{(k-1)^T} D_k^{-1} D_k^{-1} \boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T} D_k^{-1} \boldsymbol{z}^{(k-1)}}$$

or

$$\alpha_k^{\text{BB2}} = \frac{\boldsymbol{s}^{(k-1)^T} D_k \boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T} D_k D_k \boldsymbol{z}^{(k-1)}}$$

where $\boldsymbol{s}^{(k-1)} = \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right)$ and $\boldsymbol{z}^{(k-1)} = \left(\boldsymbol{g}^{(k)} - \boldsymbol{g}^{(k-1)}\right)$.

Scaled Gradient Projection (SGP) method: basic notation

Scaled Gradient Projection (SGP) method: basic notation

- Scaling matrix:

$$D_k \in \mathcal{D}_L = \{D \text{ s.p.d.} \in \mathbb{R}^{n\times n} \mid \|D\| \le L, \|D^{-1}\| \le L\}, L > 1,$$

if D_k is diagonal, the requirement leads to:

$$L^{-1} \le (D_k)_{ii} \le L.$$

Scaled Gradient Projection (SGP) method: basic notation

- **Scaling matrix**:

  $$D_k \in \mathcal{D}_L = \{D \text{ s.p.d.} \in \mathbb{R}^{n\times n} \mid \|D\| \le L, \|D^{-1}\| \le L\}, L > 1,$$

  if D_k is diagonal, the requirement leads to:

  $$L^{-1} \le (D_k)_{ii} \le L.$$

- **Projection operator**:

  $$P_{\Omega,D}(\boldsymbol{x}) \equiv \underset{\boldsymbol{y}\in\Omega}{\arg\min} \|\boldsymbol{x} - \boldsymbol{y}\|_D, \text{ where } \|\boldsymbol{x}\|_D = \sqrt{\boldsymbol{x}^T D \boldsymbol{x}}.$$

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0, 1)$, M positve integer.

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0, 1)$, M positve integer.

1. *Initialization.* Set $\boldsymbol{x}^{(0)} \in \Omega$, $D_0 \in \mathcal{D}_L$, $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0,1)$, M positve integer.

1. *Initialization.* Set $\boldsymbol{x}^{(0)} \in \Omega$, $D_0 \in \mathcal{D}_L$, $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$

For k=0,1,2,...

end

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0, 1)$, M positve integer.

1. *Initialization.* Set $\boldsymbol{x}^{(0)} \in \Omega$, $D_0 \in \mathcal{D}_L$, $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$

For k=0,1,2,...

2. *Projection.* $\boldsymbol{y}^{(k)} = P_{\Omega, D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)}))$;

   If $\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)}$ then stop.

end

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0,1)$, M positve integer.

1. *Initialization.* Set $\boldsymbol{x}^{(0)} \in \Omega$, $D_0 \in \mathcal{D}_L$, $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$

For k=0,1,2,...

2. *Projection.* $\boldsymbol{y}^{(k)} = P_{\Omega, D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)}))$;
   If $\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)}$ then stop.
3. *Descent direction.* $\boldsymbol{d}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}$.

end

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0,1)$, M positve integer.

1. *Initialization.* Set $\boldsymbol{x}^{(0)} \in \Omega, \quad D_0 \in \mathcal{D}_L, \quad \alpha_0 \in [\alpha_{min}, \alpha_{max}]$

For k=0,1,2,...

2. *Projection.* $\boldsymbol{y}^{(k)} = P_{\Omega, D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)}))$;
   If $\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)}$ then stop.
3. *Descent direction.* $\boldsymbol{d}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}$.
4. *Line-search.* Set $\lambda_k = 1$ and $\bar{f} = \max_{0 \le j \le min\{k, M-1\}} f(\boldsymbol{x}^{(k-j)})$
   While $f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}) > \bar{f} + \gamma \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}$
   $\lambda_k = \beta \lambda_k$
   end.
   Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}$.

end

Scaled Gradient Projection (SGP) method

Given $0 < \alpha_{min} < \alpha_{max}$, $\beta, \gamma \in (0, 1)$, M positve integer.

1. *Initialization.* Set $\boldsymbol{x}^{(0)} \in \Omega$, $D_0 \in \mathcal{D}_L$, $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$

For k=0,1,2,...

2. *Projection.* $\boldsymbol{y}^{(k)} = P_{\Omega, D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)}))$;
   If $\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)}$ then stop.
3. *Descent direction.* $\boldsymbol{d}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}$.
4. *Line-search.* Set $\lambda_k = 1$ and $\bar{f} = \max_{0 \le j \le min\{k, M-1\}} f(\boldsymbol{x}^{(k-j)})$
   While $f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}) > \bar{f} + \gamma \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}$
   $\lambda_k = \beta \lambda_k$
   end.
   Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}$.
5. *Update.* Define D_{k+1} and $\alpha_{k+1} \in [\alpha_{min}, \alpha_{max}]$.

end

SGP acceleration techniques

The acceleration technique involves:

SGP acceleration techniques

The acceleration technique involves:

- selection of the step-length α_k:
  general algorithm

SGP acceleration techniques

The acceleration technique involves:

- selection of the step-length α_k:
  general algorithm
- definition of the scaling matrix D_k:
  problem dependent: related to $\nabla f(\boldsymbol{x})$

SGP step-length selection

Let $\alpha_{min} = 10^{-3}, \alpha_{max} = 10^5, M_\alpha = 3, \tau = 0.5$

SGP step-length selection

Let $\alpha_{min} = 10^{-3}, \alpha_{max} = 10^5, M_\alpha = 3, \tau = 0.5$

if $\quad {s^{(k-1)}}^T D_k^{-1} z^{(k-1)} \leq 0$

$\quad \alpha_k^{BB1} = \alpha_{max}$

else

$$\alpha = \frac{{s^{(k-1)}}^T D_k^{-1} D_k^{-1} s^{(k-1)}}{{s^{(k-1)}}^T D_k^{-1} z^{(k-1)}}$$

$$\alpha_k^{BB1} = \min\{\alpha_{max}, \max\{\alpha_{min}, \alpha\}\}$$

end

if $\quad {s^{(k-1)}}^T D_k z^{(k-1)} \leq 0$

$\quad \alpha_k^{BB2} = \alpha_{max}$

else

$$\alpha = \frac{{s^{(k-1)}}^T D_k z^{(k-1)}}{{z^{(k-1)}}^T D_k D_k z^{(k-1)}}$$

$$\alpha_k^{BB2} = \min\{\alpha_{max}, \max\{\alpha_{min}, \alpha\}\}$$

end

SGP step-length selection

Let $\alpha_{min} = 10^{-3}, \alpha_{max} = 10^5, M_\alpha = 3, \tau = 0.5$

if $\quad s^{(k-1)^T} D_k^{-1} z^{(k-1)} \leq 0$

$\quad \alpha_k^{\text{BB1}} = \alpha_{max}$

else

$$\alpha = \frac{s^{(k-1)^T} D_k^{-1} D_k^{-1} s^{(k-1)}}{s^{(k-1)^T} D_k^{-1} z^{(k-1)}}$$

$\quad \alpha_k^{\text{BB1}} = \min\{\alpha_{max}, \max\{\alpha_{min}, \alpha\}\}$

end

if $\quad s^{(k-1)^T} D_k z^{(k-1)} \leq 0$

$\quad \alpha_k^{\text{BB2}} = \alpha_{max}$

else

$$\alpha = \frac{s^{(k-1)^T} D_k z^{(k-1)}}{z^{(k-1)^T} D_k D_k z^{(k-1)}}$$

$\quad \alpha_k^{\text{BB2}} = \min\{\alpha_{max}, \max\{\alpha_{min}, \alpha\}\}$

end

if $\quad \alpha_k^{\text{BB2}} / \alpha_k^{\text{BB1}} < \tau$

$\quad \alpha_k = \min\{\alpha_{k-j}^{\text{BB2}}, \quad j = 0, \ldots, M_\alpha - 1\}$

$\quad \tau = \tau * 0.9$

else

$\quad \alpha_k = \alpha_k^{\text{BB1}}$

$\quad \tau = \tau * 1.1$

end

Convergence of SGP

$$\begin{array}{c} \min f(\boldsymbol{x}) \\ \text{sub. to } \boldsymbol{x} \in \Omega \end{array} \tag{1}$$

Ω	is a convex and closed set
$f(\boldsymbol{x})$	is countinuously differentiable in Ω

Theorem

Assume that the level set $\Omega_0 = \{\boldsymbol{x} \in \Omega : f(\boldsymbol{x}) \leq f(\boldsymbol{x}^{(0)})\}$ is bounded. Every accumulation point of the sequence $\{\boldsymbol{x}^{(k)}\}$ generated by the algorithm SGP is a stationary point of (1).

Multicore/Multiprocessor architectures

Multicore/Multiprocessor architectures

Shared Memory (GPU)

Multicore/Multiprocessor architectures

Shared Memory (GPU)

Distributed Memory (PC clusters)

3D Image Deblurring

Image acquisition model:

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{n},$$

where:

- $\boldsymbol{y} \in \mathbb{R}^n$ observed image,
- $H \in \mathbb{R}^{n \times n}$ blurring operator,
- $\boldsymbol{b} \in \mathbb{R}^n$ background radiation,
- $\boldsymbol{n} \in \mathbb{R}^n$ unknown noise.

3D Image Deblurring

Image acquisition model:

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{n},$$

where:

- $\boldsymbol{y} \in \mathbb{R}^n$ observed image,
- $H \in \mathbb{R}^{n \times n}$ blurring operator,
- $\boldsymbol{b} \in \mathbb{R}^n$ background radiation,
- $\boldsymbol{n} \in \mathbb{R}^n$ unknown noise.

$$\begin{aligned} f(\boldsymbol{x}) &= D_{KL}(H\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{y}) \\ &= \sum_{i=1}^n \left(\sum_{j=1}^n H_{ij} x_j + b_i - y_i - y_i \log \frac{\sum_{j=1}^n H_{ij} x_j + b_i}{y_i} \right) \\ \Omega &= \{ \boldsymbol{x} \in \mathbb{R}^n \mid x_i \geq 0, \ i = 1, \ldots, n \} \end{aligned}$$

A suited reconstruction is obtained by early stopping the SGP iterations.

3D Image Deblurring

Image acquisition model:

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{n},$$

where:

- $\boldsymbol{y} \in \mathbb{R}^n$ observed image,
- $H \in \mathbb{R}^{n\times n}$ blurring operator,
- $\boldsymbol{b} \in \mathbb{R}^n$ background radiation,
- $\boldsymbol{n} \in \mathbb{R}^n$ unknown noise.

$$\begin{aligned} f(\boldsymbol{x}) &= D_{KL}(H\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{y}) \\ &= \sum_{i=1}^n \left(\sum_{j=1}^n H_{ij}x_j + b_i - y_i - y_i \log \frac{\sum_{j=1}^n H_{ij}x_j + b_i}{y_i} \right) \end{aligned}$$

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid x_i \geq 0,\ i = 1, \ldots, n\}$$

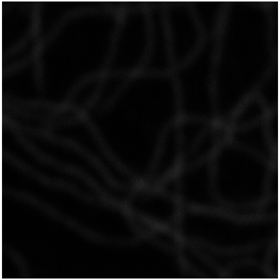
A suited reconstruction is obtained by early stopping the SGP iterations.

$$D_k = \min\left(L, \max\left(\mathrm{diag}(\boldsymbol{x}^{(k)}), L^{-1}\right)\right), \qquad L > 1$$

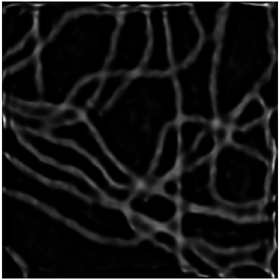
3D Image Deblurring

Electronic microscopy test example $256 \times 256 \times 52$ (betatubuline)

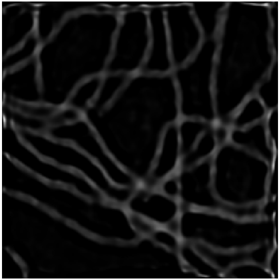
Measured Image



EM 300 it.



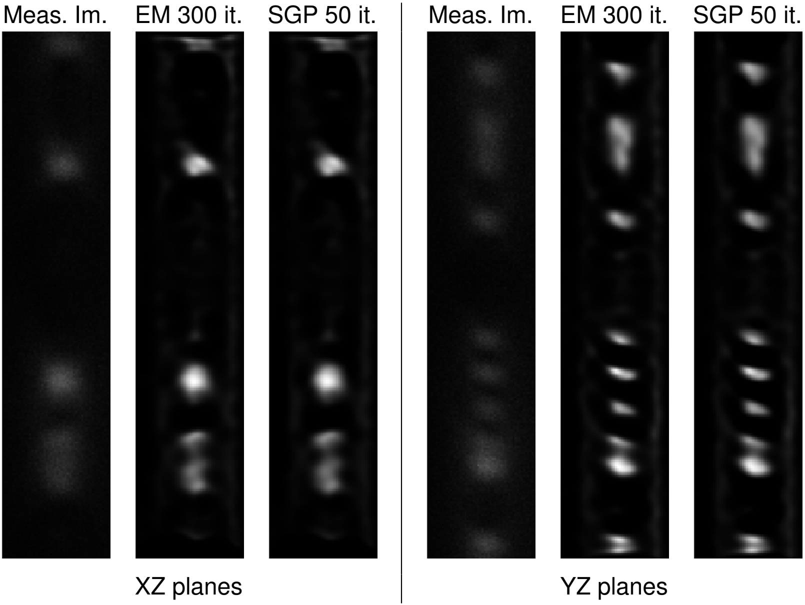
SGP 50 it.



XY planes

SGP: comparable results in fewer iterations than EM.

3D Image Deblurring



3D Image Deblurring

		SGP		*EM*	
Arch.	proc.	it	time [s]	it	time [s]
cpu	1	50	41.54	300	119.61
GPU	448	50	7.30	300	36.48
MPI	1	50	46.10	300	146.24
MPI	4	50	12.50	300	39.66
MPI	8	50	6.35	300	19.03
MPI	16	50	3.45	300	11.67
MPI	32	50	1.70	300	5.99

Image size: 256x256x52

SGP 50 it. and EM 300 it. provide comparable reconstructions:

- 3D experiments confirm that SGP provides an acceleration of EM

3D Image Deblurring

		SGP		*EM*	
Arch.	proc.	it	time [s]	it	time [s]
cpu	1	50	41.54	300	119.61
GPU	448	50	7.30	300	36.48
MPI	1	50	46.10	300	146.24
MPI	4	50	12.50	300	39.66
MPI	8	50	6.35	300	19.03
MPI	16	50	3.45	300	11.67
MPI	32	50	1.70	300	5.99

Image size: 256x256x52

SGP 50 it. and EM 300 it. provide comparable reconstructions:

- 3D experiments confirm that SGP provides an acceleration of EM
- time improvements outperform the slightly heavier iteration complexity of SGP

3D Image Deblurring

		SGP		*EM*	
Arch.	proc.	it	time [s]	it	time [s]
cpu	1	50	41.54	300	119.61
GPU	448	50	7.30	300	36.48
MPI	1	50	46.10	300	146.24
MPI	4	50	12.50	300	39.66
MPI	8	50	6.35	300	19.03
MPI	16	50	3.45	300	11.67
MPI	32	50	1.70	300	5.99

Image size: 256x256x52

SGP 50 it. and EM 300 it. provide comparable reconstructions:

- 3D experiments confirm that SGP provides an acceleration of EM
- time improvements outperform the slightly heavier iteration complexity of SGP
- both implementations show almost optimal speedup

Library Features

Main features:

- Optimization based library for solving imaging problems
- Exploits Massive Parallel Processors (GPU), or distribuited memory architectures
- Matlab source provided

Library Features

Main features:

- Optimization based library for solving imaging problems
- Exploits Massive Parallel Processors (GPU), or distribuited memory architectures
- Matlab source provided

Already tested on:

- GPU gtx 280, tesla c2050
- CINECA sp6 (IBM P575 Power 6)