**GPU implementation of a 3D bayesian CT algorithm and its application on real foam reconstruction**

Nicolas Gac¹, Alexandre Vabre², Ali Mohammad-Djafari¹, Asier Rabanal¹, Fanny Buyens²

¹ Laboratoire des Signaux et Systèmes (L2S, UMR 8506 CNRS - SUPELEC - Univ Paris Sud 11) Supélec, Plateau de Moulon, F-91192 Gif-sur-Yvette, FRANCE.
² CEA-LIST, F-91191 Gif sur Yvette, FRANCE

Emails: nicolas.gac@lss.supelec.fr, alexandre.vabre@cea.fr, djafari@lss.supelec.fr

First International Meeting on Image Formation in X-Ray Computed Tomography, 6-9 June 2010, Salt Lake City, US

---

**Few projections challenge**

**Challenge:** 3D CT cone beam reconstructions from limited projections (like in dose reduction context) require alternative methods to standard analytical filtered backprojection.

**Proposed method:** A bayesian iterative algorithm based on a Gauss/Markov/Potts model.

**Beyond limitations:** Parallelization on a 8 GPUs server has allowed us to go beyond the computing time limitations.

---

**Inverse problem:** Getting the object \( f \) from the projection data \( g \) collected from a cone beam 3D CT:

\[
g = Hf + \epsilon
\]

\(^{(1)}\)

**Prior model:** Object \( f(r) \) is composed of \( K \) regions \( \mathcal{R}_k \) corresponding to \( K \) materials labeled by a hidden variable \( z(r)=k \). A Markov/Potts model corresponding to the compactness of materials is used for \( z \).

It’s a Gaussian model corresponding to the homogeneity of materials is used for each region \( \mathcal{R}_k \). It’s:

\[
p(f(r)/z(r) = k) = N(m_k, n_k)
\]

\(^{(2)}\)

---

**Steps of the Iterative method:**

1) **Reconstruction step:** Updating \( f \) by computing \( f^{(i+1)} = \arg \max_y \{p(f|z, \theta, g)\} \). This is done by using a gradient type optimization algorithm:

\[
f^{(i+1)} = f^{(i)} + \alpha \left[ H'(g - Hf^{(i)}) + \lambda D'Df^{(i)} \right]
\]

\(^{(3)}\)

2) **Segmentation step:** Updating \( z \) by generating a sample from \( p(z|f, \theta, g) \) with a sampling algorithm from a Potts-Markov model.

3) **Characterization step:** Updating the hyperparameters using \( p(\theta|f, z, g) \). This step can be done either analytically or by sampling from known probability laws such as Gaussians or Inverse Gamma.

---

**GPU implementation of the \( H \) and \( H' \) operators**

**Goal:** Acceleration of the projection \( (Hf) \) and backprojection \( (H'g) \) which are the most time consuming operators.

**GPU acceleration:** Thanks to an implementation on Graphic Processing Unit we reach a two orders of magnitude acceleration.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projector</td>
<td>755 ms (128 ms for CPU/GPU memory transfer)</td>
</tr>
<tr>
<td>Backprojector</td>
<td>234 ms (133 ms for CPU/GPU memory transfer)</td>
</tr>
</tbody>
</table>

Reconstruction time on a GTX 295 (96 * 256³ data)

---

**Foam reconstructed (CEA-LIST real data set)**

The data set is coming from a study on water kinetics in open-cell nickel foams using x-ray microtomography. The experiments are conducted on a small sample size (1 mm³ foam) to estimate the thin geometry and model the water behavior at a scale of few pores.

---

**Future works**

- Optimization of our Gauss/Markov/Potts method
- Optimization of the CPU/GPU memory transfer
- Parallelization on the 8 GPU server of other operators (3D convolution, Potts sampling...)
- Semi automatic setting of the regularization parameters
- Technologic transfer with an industrial partner