

New GPU implementation of Separable Footprint (SF) Projector and Backprojector : first results

Camille Chapdelaine^{1,2}, Nicolas Gac¹, Ali Mohammad-Djafari¹, Estelle Parra²

¹Groupe Probl mes Inverses, Laboratoire des Signaux et Syst mes (L2S, UMR 8506 CNRS - SUPELEC - Univ Paris Sud 11)
Sup lec, Plateau de Moulon, 91192 Gif-sur-Yvette, FRANCE.

² SAFRAN SA, Safran Tech, P le Technologie du Signal et de l'Information, Rue des Jeunes Bois-Ch teaufort CS 80112,78772 Magny-Les-Hameaux, FRANCE.

camille.chapdelaine@safrangroup.com, nicolas.gac@l2s.centralesupelec.fr, Ali.Mohammad-Djafari@l2s.centralesupelec.fr, estelle.parra@safrangroup.com

The Fifth International Conference on Image Formation in X-Ray Computed Tomography, May 20-23, Fort Douglas/Olympic Village, Salt Lake City, Utah, USA

Summary: Model-based iterative reconstruction (MBIR) methods enable to improve the quality of reconstruction in 3D X-ray Computed Tomography (CT). The main computational burden of these methods lies in successive projection and backprojection operations. Among existing pairs of projector and backprojector, Separable Footprint (SF) pair combines computational efficiency and accurate modelling of X-rays passing through the volume to image. In order to accelerate these operators, implementations on Graphical Processor Units (GPUs) for parallel-computing have been proposed for SF pair. Due to a CPU-loop, these implementations involve many memory transfers between CPU and GPU which are known to be the main bottleneck for GPU computing. In this paper, we investigate a new GPU implementation of SF projector and backprojector in order to minimize these memory transfers. Our proposed GPU SF projector and backprojector have no CPU-loop, and use two ray-driven kernels for the projection and one voxel-driven kernel for the backprojection. After having described their implementations, we study these operators as single modules and validate it in a MBIR method. Perspectives for this work are GPU optimizations and comparisons with the other existing implementations of SF pair.

SF Projector

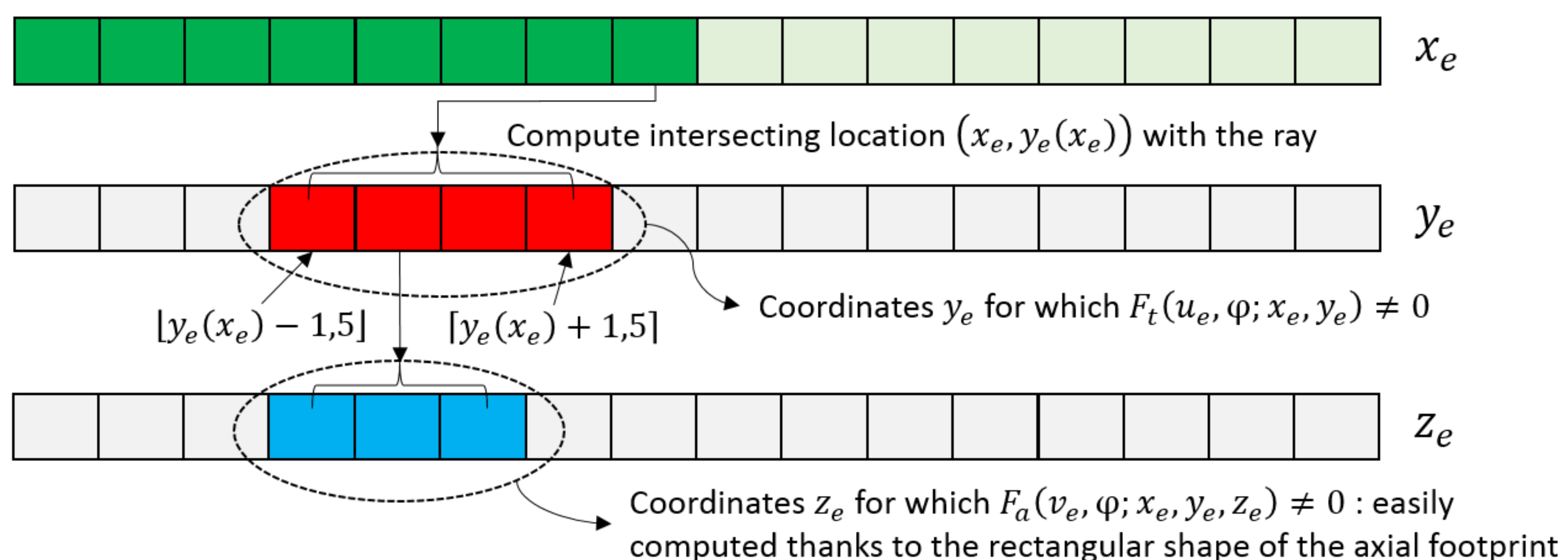
- SF projections \mathbf{g} from volume \mathbf{f} :

$$g(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) \sum_{x_e, y_e} l_{\psi_v}(\phi; x_e, y_e) F_t(u_e, \phi; x_e, y_e) \sum_{z_e} F_a(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e)$$

- F_t is trapezoidal transaxial footprint and F_a rectangular axial footprint
- One thread for one ray (u_e, v_e, ϕ) to avoid writing conflicts
- Avoid divergence between threads \Rightarrow distinguish rays according to whether the source is closer to x-axis or y-axis \Rightarrow run two independent kernels : $proj_x_ker$ and $proj_y_ker$ (in which roles of x and y are reversed)

Kernel $proj_x_ker$: SF projection for ray (u_e, v_e, ϕ) with primary direction x

for each x_e



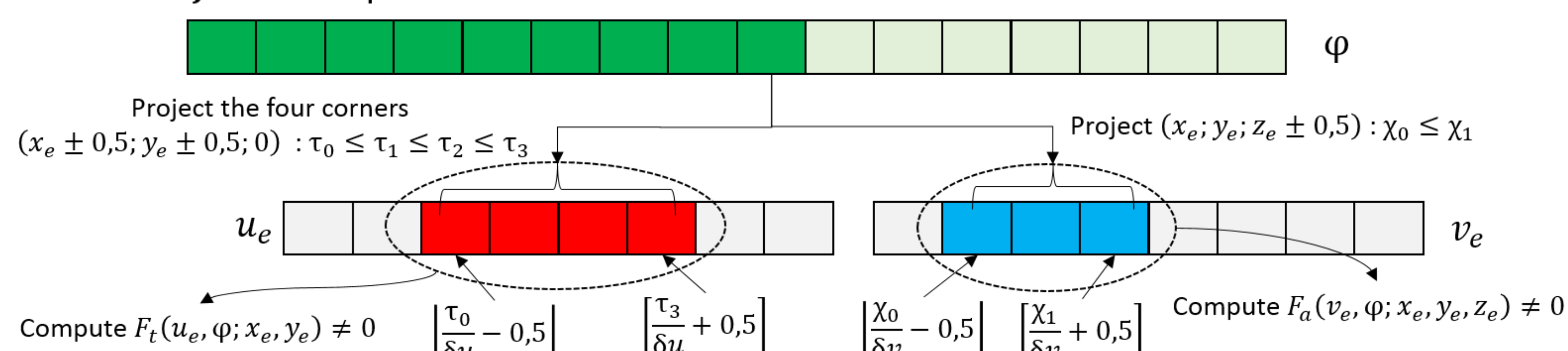
SF Backprojector

- SF backprojection \mathbf{b} from projections \mathbf{g} :

$$b(x_e, y_e, z_e) = \sum_{\phi} \sum_{u_e} F_t(u_e, \phi; x_e, y_e) l_{\psi_v}(\phi; x_e, y_e) \sum_{v_e} F_a(v_e, \phi; x_e, y_e, z_e) l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi)$$

- One thread for one voxel (x_e, y_e, z_e) to avoid writing conflicts

for each ϕ



GPU SF projector and backprojector as single modules

- First results with full volume and projections copied on the GPU
- Test on a volume with 256^3 voxels. 64 projections with 256^2 pixels are used.
- Compare with CPU version by computing the NRMSE

$$NRMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{y_i^{(GPU)} - y_i^{(CPU)}}{y_i^{(CPU)}} \right)^2}$$

Operator	Computation time		NRMSE (%)	Registers/thread
	CPU	GPU		
Projector	143.9 s	8.3 s	1.2×10^{-4}	84
Backprojector	98.7 s	4.4 s	3.2×10^{-5}	63

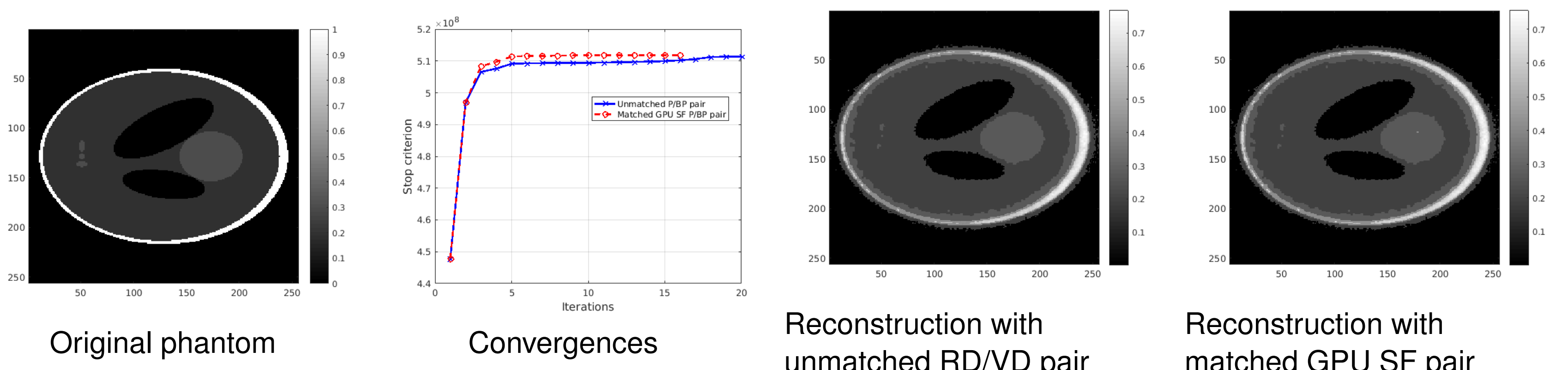
Validation in full MBIR algorithm

- Use of GPU SF pair in a joint reconstruction and segmentation algorithm derived thanks to a Bayesian approach
- Gauss-Markov-Potts prior model on the volume
- Maximize the joint posterior distribution of volume \mathbf{f} and hyperparameters θ

$$\mathcal{L}(\mathbf{f}, \theta) = \ln(p(\mathbf{g}|\mathbf{f}, \theta)) + \ln(p(\mathbf{f}|\theta)) + \ln(p(\theta))$$

- Compare results with an unmatched ray-driven/voxel-driven (RD/VD) pair

Used P/BP pair	Computation Time	\mathcal{L}_2 -relative error	Number of global iterations	Final value of the criterion
Unmatched	629.3 s	18.5 %	20	5.1136×10^8
GPU SF	6517.0 s	18.7 %	16	5.1183×10^8



Conclusion and perspectives

- Further optimizations and adaptation for very large volumes
- Comparison with other GPU implementations of SF pair