Accélération sur serveur multi-GPUs de la reconstruction 3D d'une mousse de nickel par méthodes itératives algébriques régularisées

Nicolas Gac, Alexandre Vabre, Ali Mohammad-Djafari, Fanny Buyens

Accélération sur serveur multi-GPUs de la reconstruction 3D d'une mousse de nickel par méthodes itératives algébriques régularisées

Nicolas GAC[1], Alexandre VABRE[2], Ali MOHAMMAD-DJAFARI[1],Fanny BUYENS[2]

[1]L2S, Laboratoire des Signaux et Systemes (CNRS-SUPELEC-UPS)
[2]CEA/Saclay, LIST, Laboratoire Images et Dynamique
F-91191 Gif sur Yvette, France
gac@lss.supelec.fr, Alexandre.Vabre@cea.fr, djafari@lss.supelec.fr

Résumé – Cet article traite de la reconstruction 3D en tomographie X de volume de grandes tailles (1024^3) à partir d'un nombre limité de projections. Lors d'acquisitions incomplètes en raison d'un temps limité d'acquisition ou bien dans le but de réduire la dose de rayon X, les méthodes analytiques standards de type rétroprojection filtrées offrent une qualité de reconstruction décevante. Les méthodes itératives algébriques régularisées permettent d'aller au delà de ces limitations mais souffrent d'un coût de calcul prohibitif pour son utilisation en pratique. Dans cet article, nous présentons la parallélisation des calculs des principaux opérateurs (projection, rétroprojection, convolution 3D) sur les processeurs graphiques de type "many cores". Cette parallélisation a permis une accélération significative de la reconstruction (facteur 300 sur données 1024^3 à l'aide de 8 GPUs). Par ailleurs, les résultats des reconstruction sur données simulées (phantom de Shepp Logan) et réelles en Contrôle Non Destructif (mousse de nickel) offrant une nette amélioration de la qualité de reconstruction par rapport à la méthode analytique standard FDK sont également présentés.

1 Méthode itérative des Moindres Carrés avec Régularisation Quadratique (MCRQ)

Dans la modélisation discrète et linéaire de l'acquisition CT [1], les données $\boldsymbol{g}$ provenant du scanner sont liées au volume $\boldsymbol{f}$ selon $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}$ avec $\boldsymbol{H}$ la matrice système et $\boldsymbol{\epsilon}$ les erreurs de modélisation et de mesures. La reconstruction tomographique est ainsi un problème inverse consistant à estimer $\boldsymbol{f}$ à partir de $\boldsymbol{g}$. Les méthodes analytiques de reconstruction couramment utilisées comme la méthode FDK [2] correspondent à une rétroprojection filtrée $\hat{\boldsymbol{f}}_{FRP} = (\boldsymbol{H}^t\boldsymbol{H})^{-1}\boldsymbol{H}^t\boldsymbol{g}$. Ces méthodes sont satisfaisantes dans la plupart des cas mais souffrent de nombreux artefacts de reconstruction lorsque le nombre de projections est limité. En géométrie conique pour un volume de N_{xyz}^3 voxels recontruit à partir de N_ϕ projections sur des plans de N_{uv} détecteurs, ces artefacts s'accroissent dès lors que $N_\phi < N_{xyz}$. Les méthodes algébriques permettent d'aller au delà de ces limitations en définissant la solution $\hat{\boldsymbol{f}}$ comme le minimum d'un critère. Ainsi la solution de la méthode des Moindres Carrées est définie par $\widehat{\boldsymbol{f}}_{MC} = \arg\min_{\boldsymbol{f}} \{J(\boldsymbol{f})\} = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2$. Afin de réduire la sensibilité au bruit $\boldsymbol{\epsilon}$, un terme de régularisation correspondant au Laplacien du volume est ajouté au critère dans la méthode des Moindres Carrées avec Régularisation Quadratique :

$$\widehat{\boldsymbol{f}}_{MCRQ} = \arg\min_{\boldsymbol{f}} \left\{ J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 + 2\lambda\|\boldsymbol{D}\boldsymbol{f}\|^2 \right\}$$

La paramètre de régularisation λ permet d'ajuster cet a priori de douceur et de minimiser la contribution du bruit haute fréquence $\boldsymbol{\epsilon}$ à la solution $\hat{\boldsymbol{f}}_{MCRQ}$. La solution $\hat{\boldsymbol{f}}_{MCRQ}$ peut être obtenue par un algorithme d'optimisation de descente de gradient :

$$\begin{cases} \boldsymbol{f}^{(0)} = \boldsymbol{H}^t\boldsymbol{g} \\ \boldsymbol{f}^{(i+1)} = \boldsymbol{f}^{(i)} + \alpha\nabla J(\boldsymbol{f}^{(i)}) \\ \text{avec } \nabla J(\boldsymbol{f}) = -2\boldsymbol{H}^t(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}) + \lambda\boldsymbol{D}^t\boldsymbol{D}\boldsymbol{f} \end{cases}$$

Ainsi chaque itération comporte une projection $\widehat{\boldsymbol{g}} = \boldsymbol{H}\widehat{\boldsymbol{f}}$, une rétroprojection $\delta\boldsymbol{f}_{MC} = \boldsymbol{H}^t(\boldsymbol{g} - \widehat{\boldsymbol{g}}) = Hb^t\delta g$, le calcul du terme de régularisation $\delta\boldsymbol{f}_{RQ} = \lambda\boldsymbol{D}^t\boldsymbol{D}\widehat{\boldsymbol{f}}$ afin de mettre à jour le volume $f^{n+1} = f^n + \alpha(\delta\boldsymbol{f}_{MC} + \delta\boldsymbol{f}_{RQ})$. Notons que $\boldsymbol{D}\boldsymbol{f}$ correspond à effectuer une convolution 3D avec un filtre de taille $3\times3\times3$. Par ailleurs, dans notre étude, nous effectuons le calcul du pas optimal à chaque itération (étape supplémentaire de projection $\boldsymbol{H}$) :

$$\alpha_{opt} = -\frac{\|\nabla J(\boldsymbol{f}^{(i)})\|^2}{2\|\boldsymbol{H}\nabla J(\boldsymbol{f}^{(i)})\|^2 + 2\lambda\|\boldsymbol{D}\nabla J(\boldsymbol{f}^{(i)})\|^2}$$

2 Accélération sur un serveur de calcul multi-GPUs

Afin de réduire le temps de calcul prohibitif des algorithmes itératifs sur des grands volumes (256^3 à 1024^3), nous avons parallélisé les calculs sur les processeurs graphiques de dernière génération basés sur la technologie CUDA de Nvidia. Ces processeurs dits *many cores* permettant de paralléliser les calculs sur ses centaines de coeurs de calculs sont adaptés à l'accélération du temps de reconstruction en tomographie [3]. Nous présentons ici l'accélération d'un facteur 300 obtenu lors de

la reconstruction d'un volume de taille 1024^3 sur un serveur 8 GPUs (Tesla C1060) de la société Carri. Plusieurs versions de notre implémentation GPU de l'algorithme MCRQ ont été mises en oeuvre : parallélisation sur les 240 coeurs de calcul d'un GPU de H_P et H_{RP^t} (v2) et du Laplacien discret $\boldsymbol{D}$ (v4), parallélisation sur les 8 GPUs de $\boldsymbol{H}_P$ et $\boldsymbol{H}_{RP^t}$ (v3).

2.1 Parallélisation mono GPU

Nous avons implémenté sur GPU une paire non cohérente de projection/rétroprojection : algorithme voxel-driven avec interpolation bi-linéaire pour le rétroprojecteur et algorithme ray-driven pour le projecteur (échantillonnage régulier selon les rayons traversant le volume 3D). Chacun de ces opérateurs approxime de manière différente la matrice système H, celle ci étant de taille trop importante pour etre contenu en mémoire. Ce choix de coupler un rétroprojecteur voxel-driven avec projecteur ray-driven s'explique avant tout par les facteurs d'accélération nettement inférieures obtenus à l'aide des opérateurs duaux. En effet le projecteur voxel-driven et le rétroprojecteur ray-driven souffrent d'accès à la mémoire SDRAM du GPU plus couteux en temps d'accès (nombre d'accès en écriture plus importants).

Ainsi le découpage en threads des calculs s'est effectué selon les rayons de détecteurs pour le projecteur ray-driven et selon les voxels pour le rétroprojecteur voxel-driven. La parallélisation de la convolution 3D (opérateur D de dérivés discrètes) a tout naturellement porté sur les voxels. Pour chacun de ces opérateurs, une attention toute particulière a été porté aux accès à la mémoire SDRAM. Ces accès ont été ainsi accélérés soit en chargeant les données nécessaires sur la mémoire locale dite "shared" soit en utilisant le cache 2D de texture.

2.2 Parallélisation multi GPU

Afin de gagner un facteur d'accélération supplémentaire, les calculs des opérateurs H_P et H_{RP^t} ont été distribués sur un serveur 8 GPUs (Tesla C1060) de la société Carri. Afin de pouvoir stocker dans la mémoire SDRAM de 4 Go, les données nécessaires, un découpage des données a été effectué comme illustré sur la figure 1. Pour le projecteur H_P, chaque GPU projette un demi volume réprésentant 2 Go de données (découpage en 2 selon l'axe z) sur des parties du plan de détecteurs (découpage selon l'axe vertical du plan de détecteurs et selon les angles de projections ϕ). Pour le rétroprojecteur $\boldsymbol{H}^t_{RP}$, chaque GPU rétroprojette un demi plan de détecteurs sur une partie du volume (découpage selon l'axe z).

2.3 Résultats d'accélération

Le tableau 1 compare les temps de reconstruction d'un volume 1024^3 pour différentes implémentations des opérateurs principaux de l'algorithme MCRQ sur un serveur 8 GPUs. Nous obtenons ainsi un facteur 300 d'accélération pour une implémentation sur un serveur 8 GPUs par rapport à une version

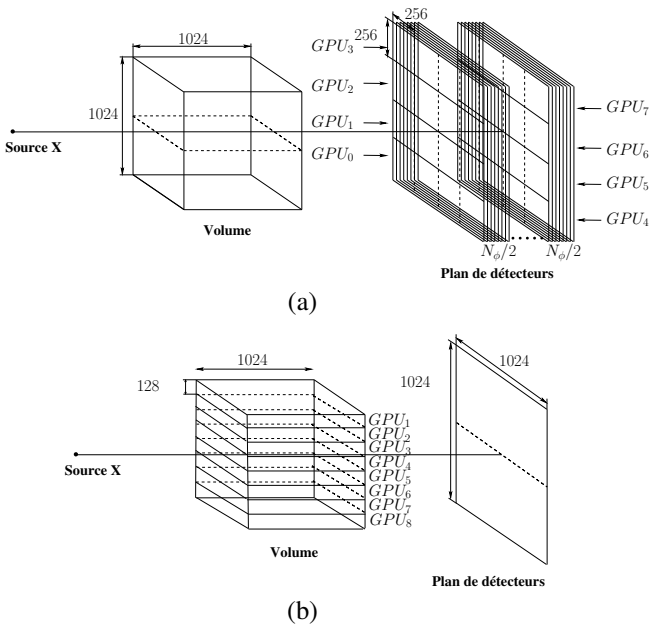


FIGURE 1 – Parallélisation multi-GPU de la projection (a) et de la rétroprojection (b).

basée sur un code CPU non optimisé. Une itération de l'algorithme MCRQ pour la reconstrcution d'un volume 1024^3 ne coute plus que 1,5 minutes au lieu de 10 heures sur une version CPU non optimisée. Nous pouvons remarqué que la parallélisation sur les GPUs a permis une accélération respective de 7 et de 5 pour la projection et la rétroprojection. Un facteur 8 d'accélération est obtenu pour la partie purement calcul, le surcout en temps de tranfert mémoire lié à la distribution des données sur les 8 GPUs explique pourquoi nous obtenons pas une parallélisation totalement efficace.

Le tableau 2 présente la coût des transfert mémoire transitant sur le PCI express entre le PC et les cartes graphiques. Nous pouvons remarquer que ces coûts sont très importants pour la convolution (68,9 %) et moyennent important pour la projection et la rétroprojection (respectivement de 37,5 % et 6,8 % sur 8 GPUs). L'utilisation des 3 canaux indépendants d'execution (chargement mémoire du PC vers le GPU, calcul sur GPU, déchargement mémoire du GPU vers le PC) des nouvelles architectures Fermi de Nvidia permettrait de masquer les transfert mémoire et de gagner jusqu'à un facteur 3 d'accélération pour ces opérateurs (cas idéal où les temps de calculs et de transfert mémoire sont équi répartis).

3 Résultats de reconstruction

3.1 Données simulées : phantom Shepp Logan

En reconstruisant des données simulant l'acquisition du phantom Shepp Logan, nous avons pu mesurés de manière quantitative l'amélioration nette de la qualité de reconstruction avec la

TABLE 1 – Temps de calcul pour une itération de la méthode MCRQ. Le volume de reconstruction est de taille 1024^3 voxels et les données correspondent à 256 projections sur des plans de 1024^2 détecteurs. Le pourcentage du temps de calcul correspondant à chaque opérateur est indiqué entre parenthèses ainsi que le facteur d'accélération obtenu après chaque optimisation.

Opérateurs	**Temps de calcul**			
	v1	v2	v3	v4
Projection $2 \times H_P$	4.1 h *(42.5 %)*	7.1 mn *(64.9 %)* $\rightarrow \times$ **35**	57 s *(21.1 %)* $\rightarrow \times$ **7**	57 s *(63.3 %)*
Rétroprojection H^t_{RP}	5.5 h *(56.9 %)*	21.8 s *(3.3 %)* $\rightarrow \times$ **908**	4.0 s *(1.5 %)* $\rightarrow \times$ **5**	4.0 s *(4.4 %)*
Convolution $3 \times D$	3.2 mn *(0.6 %)*	3.2 mn *(29.2 %)*	3.2 mn *(71.1 %)*	12.1 s *(13.4 %)* $\rightarrow \times$ **16**
Autre	17 s *(0.0 %)*	17 s *(2.6 %)*	17 s *(6.3 %)*	17 s *(18.9 %)*
Total	9.7 h	10.9 mn $\rightarrow \times$ **53**	4.5 mn $\rightarrow \times$ **2.4**	1.5 mn $\rightarrow \times$ **3.0**

v1 : H_P, H^t_{RP} et D sur CPU
v2 : H_P et H^t_{RP} sur 1 GPU, D sur CPU
v3 : H_P et H^t_{RP} sur 8 GPUs, D sur CPU
v4 : H_P et H^t_{RP} sur 8 GPUs, D sur 1 GPU

TABLE 2 – Proportion du temps de traitement consacré au transfert mémoire entre le PC et la carte graphique pour chaque opérateur lors de la reconstruction d'un volume de 1024^3 à partir de 256 projections.

	1 GPU	**8 GPUs**
Projecteur H_P	10 %	37.5 %
Rétroprojecteur H^t_{RP}	1.4 %	6.8 %
Convolution D	68.9 %	

méthode MCRQ comparée à celle des méthodes directe FDK ou itérative MC sans régularisation pour un faible nombre de projection $N_\phi = N_{xyz}/4$. Nous présentons ainsi sur la figure 2 en (a) la distance L1 entre le volume reconstruit et le volume d'origine et en (b) le profil d'une coupe du volume. Nous observons ainsi que la régularisation avec un facteur de régularisation λ correspondant comme attendu à $\frac{\sigma_\epsilon^2}{\sigma_f^2}$ permet d'éviter la divergence provoquée par le bruit de mesure ϵ tout en conservant les contours du volume.

3.2 Données réelles : mousse de nickel

Nous avons appliquées notre méthode sur des données réelles de mousse de nickel (pores de l'ordre de 500 μm) de la société INCOFOAM générées par le banc de microtomographie du CEA-LIST. Les mousses solides (os, bois, corail...) sont une classe de matériaux présentant une structure interne très poreuse donc trés légère, mais néanmoins très résistante [4]. La prédiction du comportement mécanique, thermique ou de la propagation de fluides dans ces structures est un enjeu dans de

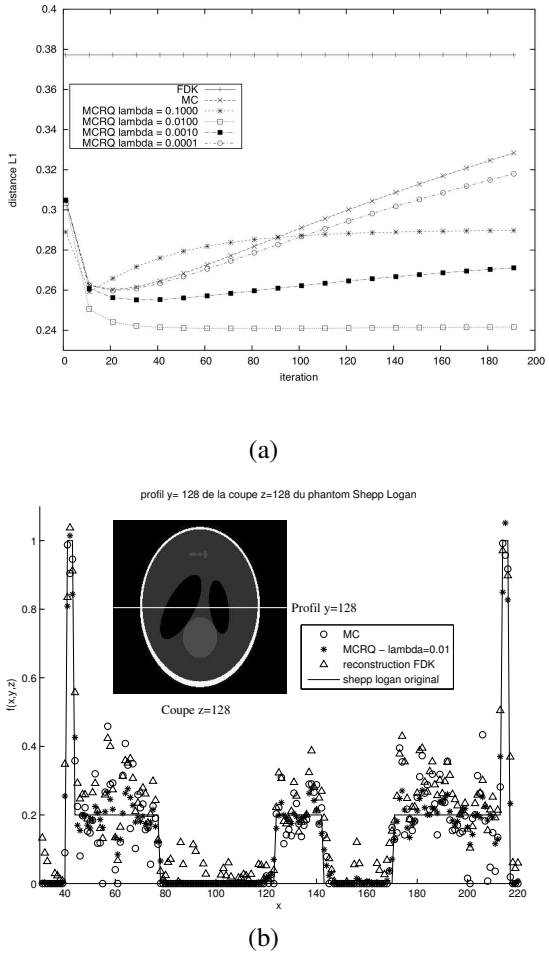
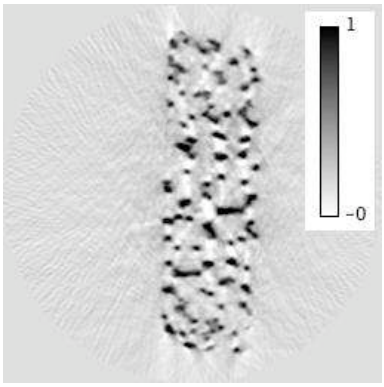


FIGURE 2 – Reconstruction d'un phantom Shepp Logan 256^3 à partir de 64 projections sur le plan de 256^2 détecteurs avec l'ajout d'un bruit de 20dB : (a) Distance L1 entre le volume reconstruit et le volume d'origine pour les méthodes FDK, MC et MCRQ au cours des itérations ; (b) Profil d'une coupe du volume reconstruit par les méthodes FDK, MC et MCRQ.

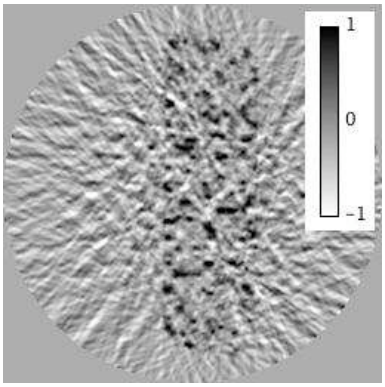
nombreux domaines (énergie, transport, sciences de matériaux) et peuvent amener à réduire le nombre de projections lors de l'acquisition. Les images de la figure 3 illustrent l'amélioration qualitative nette de la reconstruction MCQR (c) par rapport à la méthode FDK (b) lorsque le nombre de projections est faible $N_\phi = N_{xyz}/8$ (qualité comparable à la méthode FDK avec toutes les projections (a)).

4 Conclusion

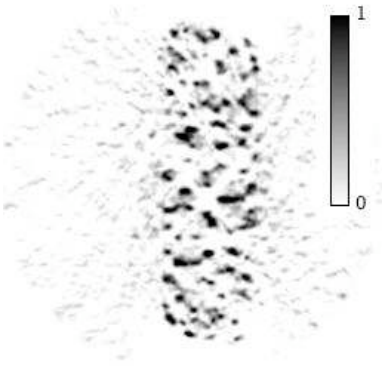
Dans cet article, le problème de la reconstruction 3D de grand volume à partir d'un nombre limité de données a été présenté. La méthode itérative avec régularisation MCRQ a prouvé de manière quantitative sur des données simulées et de manière



(a)



(b)



(c)

FIGURE 3 – Reconstructions d'une mousse 256^3 à partir de N projections sur le plan 256^2 de détecteurs : méthode directe FDK avec N=256 (a) et N=32 (b) ; méthode MCRQ avec N=32 (c).

qualitative sur des données réelles qu'elle permettait d'améliorer nettement la qualité de reconstruction. La parallélisation de cette méthode sur un serveur 8 GPUs de MCRQ a été présentée. La réduction du temps de calcul de deux ordres de grandeur ainsi obtenue permettra d'explorer d'autres méthodes de régularisation adaptées au problème du contrôle non destructif sur des données réelles.

Références

[1] A. Mohammad-Djafari, Ed., *Inverse Problems in Vision and 3D Tomography*, ISTE, 2009.

[2] L.A. Feldkamp, Davis L C, and Kress J W, "Practical cone-beam algorithm," *J Opt Soc Am*, vol. A6, pp. 612–619, 1984.

[3] N. Gac et al, "High speed 3D tomography on CPU, GPU and FPGA," *EURASIP Journal on Embedded systems*, 2008.

[4] O. Gerbaux et al, "Transport properties of real metallic foams," *J Colloid & Interface Science*, vol. 342, pp. 155–165, 2010.