Application d'un cache 2D prédictif à l'accélération de la rétroprojection TEP 2D

2D PET backprojection acceleration through a 2D predictive cache

Stéphane Mancini [1], Nicolas Gac [1], Michel Desvignes [1], Olivier Bourrion [2], Olivier Rosseto [2]

[1] Laboratoire des Images et des Signaux, Grenoble, France.
[2] Laboratoire de Physique Subatomique et Cosmologie, Grenoble, France.

Manuscrit reçu le 14 octobre 2005

Résumé et mots clés

Le développement et la diffusion des équipements TEP passent par la réduction des temps de calcul de la reconstruction des images acquises. Aussi cet article présente une solution mixte logicielle/matérielle pour l'accélération de la reconstruction 2D sur une plateforme SOPC (System on Programmable Chip), la nouvelle génération de circuits reconfigurables. Le verrou technologique posé par la latence des accès mémoire est levé grâce au cache 2D Adaptatif et Prédictif (cache 2D-AP).

Imagerie TEP, Reconstruction, Rétro-projection, Adéquation Algorithme Architecture, Partitionnement matériel/logiciel, System on Programmable Chip, Cache, FPGA.

Abstract and key words

Reduction of image reconstruction time is a key point for the development and spreading of PET scans. Thus this article presentes a hardware/software architecture which aims at accelerating the 2D reconstruction on a SoPC (System on Programmable Chip) plateform, the new generation of reconfigurable chip. Issue posed by the latency of memory accesses has been solved thanks to the 2D Aptative and Predictive cache (2D-AP cache).

PET imaging, Reconstruction, Backprojection, Algorithm Architecture Adequacy, Hardware/software partitionning, System on Programmable Chip, Cache, FPGA.

Remerciements

Nous tenons à remercier toutes les personnes qui ont contribué à ce projet et particulièrement Nicolas Costes et Anthonin Reilhac, du CERMEP, à Lyon, pour nous avoir fournis des données TEP.

1. Introduction

La tomographie d'émission de positons (TEP) est une imagerie nucléaire fonctionnelle *in vivo* utilisant des marqueurs radioactifs qui permet de nombreuses explorations des systèmes biologiques en raison de la variété des traceurs [6], complémentaire aux autres techniques d'imagerie médicale (IRM, Tomodensitométrie X, etc…). Cette imagerie consiste dans un premier temps à détecter en coïncidence les deux photons émis dans des directions opposées après annihilation du positon émis lors du processus de désintégration du traceur. Puis, la reconstruction donne une estimation de la densité volumique du traceur qui est une mesure de l'activité biologique associée au traceur. De nombreux progrès ont été réalisés dans le domaine de l'imagerie nucléaire mais la résolution et la rapidité de la reconstruction logicielle des images restent un frein majeur pour de nombreuses applications. Une reconstruction rapide, interactive, et de bonne qualité est d'un intérêt essentiel pour les nombreuses applications de l'imagerie TEP mais reste difficile à atteindre avec les systèmes actuels. Dans le domaine clinique, cela offre la possibilité de repositionner le patient si besoin ou de diminuer les doses injectées à qualité égale. Dans le domaine économique, cette amélioration réduit les coûts grâce à une meilleure disponibilité du matériel. Dans le domaine de la recherche, les études pharmacologiques sur le petit animal (micro TEP) dans de grandes séries nécessitent cette rapidité.

L'acquisition des données et la reconstruction d'image sont généralement effectuées en deux étapes distinctes, en ligne pour l'acquisition, hors ligne pour la reconstruction. Les données acquises sont des projections ou sinogrammes des données physiologiques et la reconstruction d'images ou tomographie consiste à reconstruire l'image à partir de ces projections. La notion de tomographie se retrouve également en imagerie X avec les scanners CT ou en imagerie mono photonique (SPECT). Deux grandes classes de méthodes de tomographie existent :

- les méthodes analytiques sont principalement basées sur la rétro-projection filtrée et sont utilisées dans les dispositifs commerciaux. Elles offrent un compromis actuellement acceptable entre le temps de calcul et la qualité d'image en terme de rapport signal à bruit et l'introduction des artefacts de reconstruction.
- les méthodes algébriques et probabilistes modélisent le problème par un système linéaire de grande dimension, dont la solution est obtenue de ce fait par une méthode itérative. Elles ont l'avantage de prendre en compte les phénomènes altérant les hypothèses de base des méthodes analytiques (diffusion, coïncidences fortuites, etc..) et améliore ainsi la qualité des images reconstruites, au prix d'un coût de calcul important.

Nous nous sommes intéressés dans le cadre de cet article à l'algorithme classique de rétro-projection, de la classe des méthodes analytiques, pour sa diffusion actuelle sur les différentes modalités (X, PET, SPECT) et sa maturité. Il est à la base de la reconstruction en 2D ou en 3D. Les méthodes itératives intègrent une étape de rétro-projection filtrée, dont le coût algorithmique est important dans le coût total. Il en existe de nombreuses implémentations sur du matériel dédié, sur des machines vectorielles, sur des grappes de PC ou purement software. Notre objectif est de démontrer l'intérêt des plates formes SOPC dans ce cadre. L'analyse du problème montre que la difficulté réside principalement dans la stratégie d'accès aux données, non seulement de part le volume de données à utiliser mais aussi du fait de leur accès non séquentiel dans une structure 2D. Nous avons donc cherché une solution efficace à ce goulot d'étranglement sur une plate forme de test SOPC pour un algorithme de reconstruction 2D. Dans une première partie, nous présentons un bilan de ces différentes approches pour la reconstruction par rétro-projection filtrée en TEP ou en Imagerie X, en 2D ou en 3D. Nous résumons rapidement l'algorithme de rétro-projection filtrée et les points clés de ce dernier. Les solutions adaptées sont alors proposées avant d'être évaluées.

État de l'art De nombreuses solutions ont déjà été étudiées pour accélérer les temps d'acquisitions et de reconstruction. Le problème essentiel de la reconstruction est celui de l'accès aux données car la localité spatiale en adresse et temporelle des données (sinogramme) utilisées par la rétro-projection n'est en général pas assurée. En effet, lorsqu'une donnée à l'adresse mémoire N est utilisée, la donnée suivante qui sera utilisée n'est pas à l'adresse $N+1$, et souvent, pas dans la même page, ce qui génère des défauts de cache et ralentit le processeur.

Les grandes voies étudiées sont l'optimisation algorithmique, la vectorisation ou parallélisation sur des machines spécifiques ou des grappes de calcul, les approches matérielles utilisant des composants de type DSP ou plus récemment des cartes graphiques commerciales.

D'un point de vue algorithmique, en 2D, toutes les implémentations de la rétro-projection ont une complexité en $O(N^3)$, où N est la taille de l'image à reconstruire. Une version récursive [1] exploite une approche hiérarchique et itérative qui divise le problème en sous problèmes de tailles inférieures plus simples à résoudre et utilisant une interpolation. Bien que performantes, ces méthodes introduisent des artefacts. Elles ne peuvent cependant prendre en compte les nombreuses géométries réelles rencontrées en pratiques [11]. Dans le cas tridimensionnel, les données d'entrées sont des sinogrammes 2D complets non disponibles dans le cadre de la TEP. La structure hiérarchique fortement non aléatoire complexifie l'accès aux données et le gain réel est de l'ordre de 50 %, avec un choix des paramètres difficile et très dépendant des images elles mêmes [15].

La vectorisation ou la parallélisation de la reconstruction a été et est un champ de recherche toujours très actif. Les machines vectorielles présentent l'avantage de traiter avec une seule instruction (donc un seul cycle de chargement/décodage) un ensemble de données bien localisées, au sein d'un même tableau. Quelques tentatives pour les exploiter en TEP ont été faites [8]. Elles présentent l'inconvénient d'être peu évolutives, chères et peu nombreuses. Le principe dominant actuel dans la catégorie SIMD est le Symmetric Multiprocessing ou SMP. Le

traitement des données est distribué sur plusieurs nœuds de calcul, permettant de répartir la charge de calcul. Cette architecture est efficace lorsque les opérations réalisées en parallèle sont indépendantes et coûteuses en temps devant les transmissions. Le volume des transactions ou échanges doit être limité, sous peine de voir le canal de transmission saturé et réduire considérablement les performances. Ces nœuds de calculs sont localisés soit sur une même machine et reliés par un bus (machine multi processeur) soit distribués sur des machines distantes reliées par un réseau (grappes de PC).

Dans le cas des machines multi-processeur, le nombre de nœuds de calcul est limité, le coût de communication moyen, la mémoire est généralement partagée. Notons que la rapidité des bus n'a pas suivi la loi de Moore des processeurs et que ce décalage croissant exacerbe les problèmes d'accès mémoires. Un des problèmes majeurs, l'accès aux données, n'est pas correctement pris en compte dans ce schéma et la nature généraliste de cette architecture et des processeurs associés limitent alors les gains de rapidité [19].

Du fait de leur très bon rapport prix/performance, la reconstruction sur des grappes de PC a fait l'objet de nombreux travaux [17, 10]. L'avantage de cette architecture est l'ouverture vers un nombre élevé de nœuds de calcul, potentiellement hétérogène. Le problème majeur est celui de l'accès aux données, le lien de communication étant souvent physiquement un réseau informatique classique qui limite alors les performances [21]. Sur les clusters, les communications sont souvent réalisées par passage de messages à l'aide de bibliothèques telles que PVM, MPI ou autres. La diffusion d'une donnée se fait par protocole de type unicast, une donnée est transmise à un nœud unique. Dans le cadre de la reconstruction, une donnée doit souvent être utilisée par plusieurs nœuds de calcul et les protocoles unicast génèrent un trafic de contrôle inutile. Bien que [18] ait montré une faible efficacité des broadcast MPI, [2] propose une reconstruction multicast efficace en UDP sur un faible nombre de nœuds (8). Augmenter le nombre de nœuds pose alors le problème de la gestion des voxels reconstruits et du trafic généré. D'une manière générale, les implémentations vectorielles ou parallèles classiques n'ont pas résolu le problème de l'accès aux données.

Une solution originale a exploité ces dernières années les cartes graphiques standards de nos PC de bureau. L'idée de base est d'utiliser le pipeline graphique haute performances inclus dans ces cartes [13]. Les premières implémentations avaient de fortes limitations, due à l'arithmétique entière 8 bit de ces cartes, au manque de programmation interne au pipeline. Les nouvelles générations introduisent la possibilité de réaliser des opérations sur des tableaux en les programmant astucieusement [22]. Le principal problème reste la taille mémoire disponible sur les cartes, les communications mémoire/carte limitant alors les gains.

Les solutions matérielles de type ASIC (TeraRecon) ont été implanté sur des machines commerciales, en général sur les scanners CT de part de marché plus large que les scanners PET ou SPECT. [14, 9] sont des exemples de circuits d'acquisition. La difficulté de conception et le coût de ces dispositifs les rendent peu réactifs aux avancées récentes et les développements sur ces cibles technologiques sont donc réduits actuellement. Les solutions basées sur les DSP [16] sont efficaces mais présentent les mêmes limitations en puissance de calcul que les processeurs classiques, et le problème de l'accès a des données 2D reste complexe [7]. Au contraire, les FPGA, de part leur flexibilité, leur reconfigurabilité et les performances actuelles, sont un champ très actif actuellement. Leur première application en reconstruction concerne l'acquisition des données [23], mais l'évolution technologique permet aujourd'hui d'envisager de porter les algorithmes de reconstruction sur FPGA [12, 4].

L'association de FPGA et de processeurs à usage généraux sur les plates formes SOPC ouvre une voix prometteuse pour l'implantation et la maintenance des algorithmes modernes de reconstruction tomographique. Les performances temporelles attendues sont proches des circuits dédiés, tout en conservant en grande partie la flexibilité du logiciel. Les points essentiels à prendre en compte lors de l'implantation sont les accès non séquentiels et multiples aux données, c'est-à-dire, les problèmes de localité spatiale et temporelle.

2. Objectifs

2.1. L'algorithme

L'algorithme implémenté effectue la rétro-projection des données acquises par le scanner. Ces dernières, appelées sinogrammes, forment la transformée de Radon de la fonction f représentant la densité du taux d'émission radioactive du volume observé. Le sinogramme est une matrice image à deux dimensions à K colonnes. Chaque colonne k correspond à la projection orthogonale de f sur r détecteurs placés orthogonalement sur une ligne inclinée de l'angle $\theta_k = \frac{k * \pi}{K}$ par rapport à l'axe x. Ainsi, le point du sinogramme $S(\theta_k, r)$ est la somme des pixels sur une Ligne De Réponse (LDR) du scanner qui est perpendiculaire à l'axe des détecteurs et passe par le détecteur r. À partir de ce sinogramme, l'algorithme reconstruit l'image f^* en rétro-projetant les K lignes du sinogramme dans l'espace image.

$$f^*(x,y) = \sum_{k=0}^{K} S(k, r_k(x,y)) \tag{1}$$

$$r_k(x,y) = x \cos\left(\frac{k\pi}{K}\right) - y \sin\left(\frac{k\pi}{K}\right) + \text{offset} \tag{2}$$

Cette méthode n'est pas l'exacte transformée de Radon inverse. En effet, elle produit des artefacts en étoile et l'image reconstruite devient floue. Pour améliorer la reconstruction, on peut filtrer le sinogramme mais cette étape étant indépendante de la rétro-projection 2D, nous ne la développerons pas dans cet article.

2.2. Le « challenge »

Le sinogramme étant stocké en mémoire externe de type SDRAM, nous devons avoir une gestion efficace de la mémoire pour compenser la latence et permettre un haut degré de parallélisme. Cette stratégie diffère de celle de [3] qui utilise plusieurs bancs de mémoires SRAM indépendants avec une latence nulle pour améliorer le débit mémoire. Cette dernière constitue cependant une solution coûteuse et il est possible d'obtenir de meilleures performances avec des mémoires SDRAM à faible coût malgré leur latence importante.
Les caches standards ayant pour principe un accès linéaire à la mémoire ne peuvent être une solution satisfaisante étant donné leur complexité technologique et leur faible taux de charge de données utiles. En effet, les accès mémoires nécessaires pour reconstruire un pixel image $f^*(x,y)$ suivent une sinusoïde dans le sinogramme ce qui constitue une faible localité spatiale pour les adresses mémoire. Pour accélérer ces accès mémoire, un nouveau mécanisme de cache est nécessaire. Une prédiction des points du sinogramme dont a besoin l'unité de calcul permettra au cache de charger les données pendant le processus de calcul.

2.3. Localité temporelle et spatiale 2D

Reconstruction par blocs La localité spatiale 2D et temporelle exploitée par le cache 2D adaptatif et prédictif est créée en reconstruisant l'image f^* par blocs et en modifiant l'enchaînement des boucles de l'algorithme, le sinogramme étant parcouru en phase. La reconstruction par blocs nous permet d'augmenter le taux de réutilisation des données du sinogramme selon r à k constant, ce qui est favorable à l'utilisation d'un cache. La taille du cache, qui correspond à la durée d'utilisation des données, est réduite en accroissant la localité en k.
L'ensemble des données nécessaires à la reconstruction d'un bloc de f^* est l'union des sinusoïdes nécessaires à la reconstruction de chacun des pixels du bloc. Ces données forment un « tube » de données voisines du fait de la continuité de la fonction qui associe à chaque pixel de f^* l'ensemble de coordonnées $(k,r_k(x,y)),k \in [0,\pi[$. Lors de la reconstruction d'un bloc B, pour un k donné, l'ensemble des données $(k,r_k(x,y)),(x,y) \in B$ correspond à la projection du bloc B sur la ligne de détecteur d'angle θ_k.
Dans le cas particulier de la reconstruction d'un bloc carré de taille $n*n$, B_n, $(k,r_k(x,y)),(x,y) \in B_n$ est de hauteur maximale $\sqrt{2}n$ et minimale n. En moyenne, une donnée du sinogramme est donc réutilisée au mieux n fois et $\frac{n}{\sqrt{2}}$ au minimum. Le premier cas correspond à $\theta_k \in \left\{0,\frac{\pi}{2}\right\}$, le second à $\theta_k \in \left\{\frac{\pi}{4},3\frac{\pi}{4}\right\}$. Dans ce dernier cas, au pire une donnée sinogramme est utilisée une fois et $\sqrt{2}n$ au mieux (c'est la projection d'un carré sur sa diagonale). Bien entendu, toute forme de bloc qui permet un pavage de f^* convient (par exemple des hexagones, etc ...).

L'algorithme de reconstruction est maintenant le suivant :

Pour chaque bloc B de f^*
$\forall (x,y) \in B, f^*(x,y) = 0$
$\forall k \in \{0,K-1\}$
$\forall (x,y) \in B, f^*(x,y) += S(k,r_k(x,y))$

Compromis localité/ressources Nous avons donc à trouver un compromis sur n dont l'augmentation d'une part améliore l'efficacité du cache mais accroît d'autre part les ressources matérielles du cache et la quantité de mémoire nécessaire aux variables intermédiaires. En effet, la hauteur du sinogramme mémorisée en cache évolue avec $\sqrt{2}n$ et il faut stocker les n^2 valeurs intermédiaires de $f^*(x,y)$ pour $(x,y) \in B$.
La valeur optimale de n dépend aussi des performances du cache qui sont fonctions, entre autre, des dimensions de la zone en cache, du débit, et de la latence des accès à la mémoire principale, comme nous le verrons au chapitre 3.1.

3. Architecture

3.1. Objectifs du cache 2D-AP

Le cache 2D-AP est l'élément original de l'architecture proposée en permettant aux unités de calculs d'accéder aux données sans temps mort. Ceci est rendu possible grâce à un mécanisme générique de prédiction des prochaines données utilisées. Le cache 2D-AP anticipe les accès au sinogramme 2D par une analyse de la séquence des coordonnées de pixels requis par l'unité de calcul pour prédire les prochains accès à partir d'une hypothèse à vitesse de déplacement constante sur l'image (les données), ce qui est le cas de notre application. La reconstruction d'un bloc de pixels permet de produire une séquence d'accès au sinogramme qui tire partie au mieux des caractéristiques du cache 2D-AP.

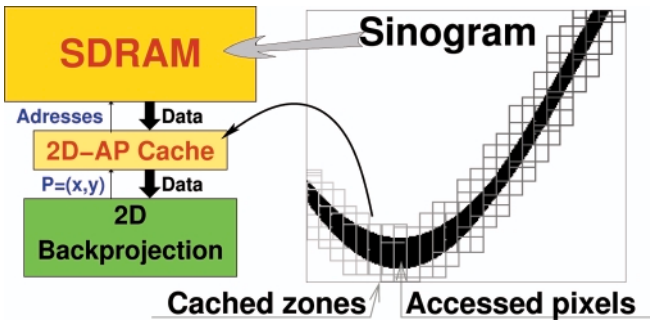


Figure 1. Concept du cache 2D-AP.

3.1.2. Fonctionnement

La prédiction de la zone en cache permet de réduire le taux de défaut de cache lorsque les accès pixels suivant se déplacent sur

l'image. Le centre et la taille de la zone sont déterminés par le calcul de la moyenne et du pseudo écart type (PSD = pseudo standard deviation) des accès pixels. La moyenne et le PSD sont calculés à l'aide de filtres passe-bas récursifs du premier ordre. En supposant une distribution uniforme des accès pixels autour de la moyenne, nous pouvons estimer que, pour une courte durée, ils sont dans un rectangle centré sur la moyenne et dont les demi-largeurs valent deux fois le PSD sur chaque axe. La zone ainsi calculée est mise à jour lorsque la moyenne varie.
Un mécanisme de prédiction permet d'anticiper la position du centre du cache lorsque les accès pixels suivent un chemin complexe dans l'image et réduit ainsi le coût de la latence des accès mémoire. La mise à jour ne se produit que lorsque la moyenne a suffisamment variée et la nouvelle zone est centrée sur une anticipation de la moyenne, supposée se déplacer à vitesse constante. Ainsi, une zone de garde est définie autour du centre de la zone en cache et la mise à jour se fait lorsque la moyenne calculée sort de cette zone de garde, dans la direction de sortie, comme illustrée figure 2.

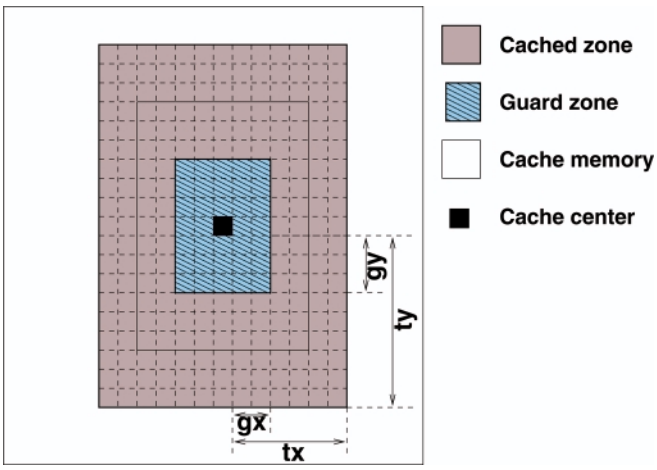


Figure 2. Zones du cache 2D-AP.

Les principaux paramètres du cache sont les fréquences de coupure nécessaires aux calculs de la moyenne et du PSD, $a = (a_x, a_y)$ et $b = (b_x, b_y)$, ainsi que les vitesses de déplacement du mécanisme de prédiction, $\alpha = (\alpha_x, \alpha_y)$. Nous avons aussi les paramètres de réglage des dimensions des différentes zones du cache : les facteurs de proportionnalité de la zone de garde, $\gamma = (\gamma_x, \gamma_y)$, de la taille du cache, $\tau = (\tau_x, \tau_y)$ et de la vitesse du cache $\alpha = (\alpha_x, \alpha_y)$.
L'état du cache, à l'instant n, est déterminé par les variables suivantes :
- $P = (x_n, y_n)$, coordonnées du point utilisé par l'unité de traitement
- $P_f = \mathrm{f}_a(P) = (x_{f_n}, y_{f_n})$, moyenne des points utilisés
- $d = |P - P^f|$, PSD courant (distance du point à la moyenne calculée)
- $d_f = \mathrm{f}_b(d)$, PSD moyen
- c, centre courant du cache
- v, PSD courant du cache. Il définit :
  - $g = \gamma v$, taille de la zone de garde
  - $t = \tau v$, taille de la zone en cache
  - $\Delta = \alpha v$, le déplacement du cache

Les paramètres v, g et t et Δ sont remis à jour à chaque déplacement du cache.
- $e = P - c$, distance du point courant au centre du cache

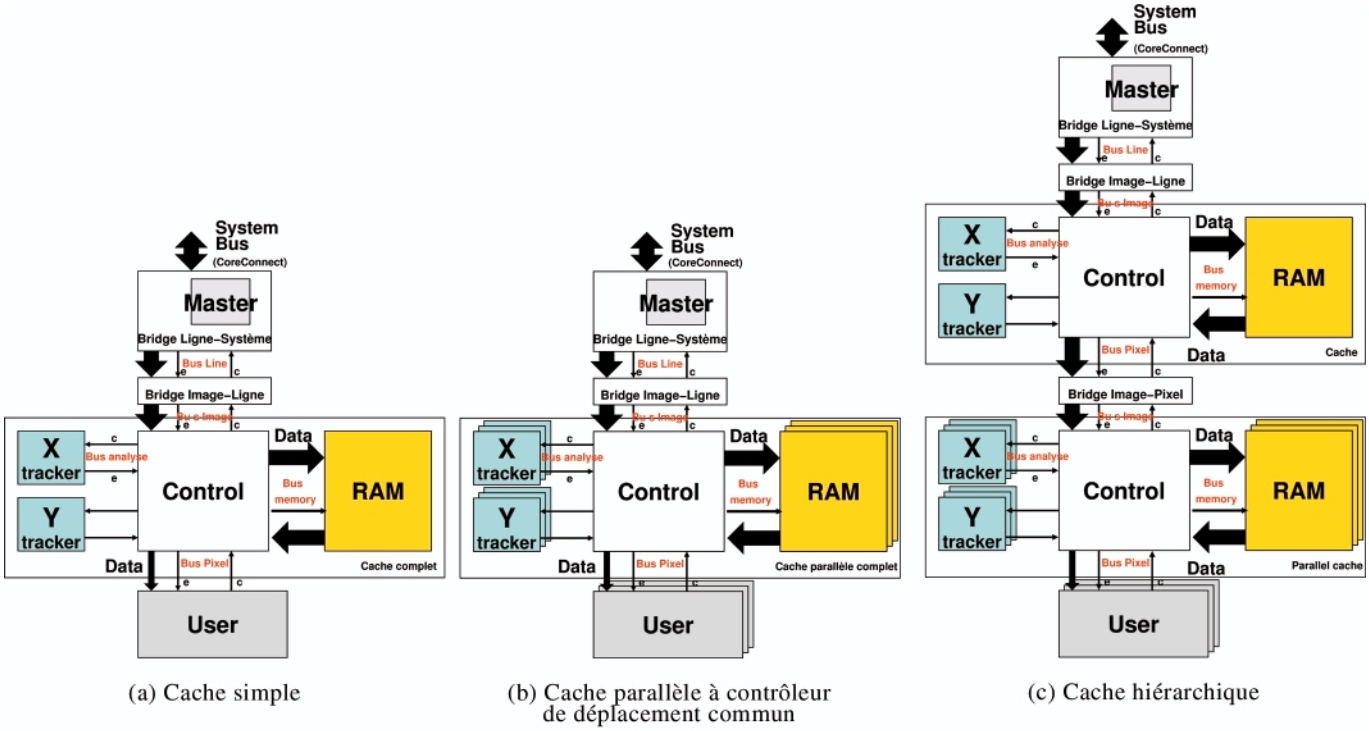
Le bloc mémoire du cache 2D-AP a une taille fixe et définit la valeur maximum de t. À chaque déplacement, seule la nouvelle partie de l'image en cache est chargée, celle étant à l'extérieur est écrasée par les nouvelles données. En effet, la position des pixels de la zone dans la mémoire est calculée modulo la taille maximum. Un défaut de cache se produit lorsqu'un pixel est hors de la zone en cache ($e > t$) et seul le pixel en défaut est chargé depuis la mémoire externe.
Pour réduire le coût matériel du contrôle et obtenir de bonnes performances les filtres passes bas pour le calcul des moyennes sont réalisés à l'aide de simples additions et décalages. Ce sont des filtres RII du premier ordre dont l'équation est $s_n = ce_n + (1-c)s_{n-1}$, e signal d'entrée et s signal filtré, et nous notons $s = \mathrm{f}^c(e)$. Le paramètre $1-c$ correspond à la constante de temps du filtre RC équivalent. Lorsque les suites s et e sont représentées en virgule fixe et c une puissance de $\frac{1}{2}$, nous ne réalisons plus que des additions et décalages.

3.1.3. Architecture

L'architecture proposée permet d'une part de construire rapidement un cache grâce à la modularité du système par assemblage de composants et d'autre part la conception d'une hiérarchie mémoire complexe qui permet d'alimenter une partie opérative à partir de plusieurs niveaux de cache successifs.

Une architecture modulaire L'architecture modulaire du cache, représentée figure 3(a), permet de construire rapidement une hiérarchie mémoire complexe. Le système est décomposé en éléments qui s'interfacent par des bus de transfert d'imagettes de l'image lue, de lignes et de pixels, ce dernier bus étant l'interface utilisateur. Le bus connecté aux analyseurs (« tracker » sur la figure 3(a)) propose un protocole unifié qui permet de modifier le type d'analyseur pourvu que ce protocole soit respecté. Ainsi, nous disposons d'analyseurs de différents type (estimation de la moyenne et de la variance, estimation de la seule moyenne, ou suiveur linéaire) dont le choix dépend de l'application et du niveau de hiérarchie.

Un modèle hiérarchique Une hiérarchie mémoire est construite en alimentant un cache de niveau inférieur à partir d'un cache de niveau supérieur, ainsi qu'illustré figure 3(c). Le cache parallèle, représenté seul figure 3(b), gère des accès en parallèle et le bus image est partagé à travers un contrôleur de déplacement commun, ce dernier réalisant l'arbitrage. Les requêtes de mise à jour du cache inférieur sont transformées en accès au cache supérieur. Il est également possible de construire un système pour lequel plusieurs caches, avec chacun leur



(a) Cache simple (b) Cache parallèle à contrôleur de déplacement commun (c) Cache hiérarchique

Figure 3. Architectures du cache 2D-AP hiérarchique.

contrôleur de déplacement, partagent un bus image doté d'un arbitre.

3.2. Analyse du cache et méthode de réglage

3.2.1. Modèle de coût de déplacement

La durée d'une mise à jour du cache dépend du sens dans lequel celui-ci se déplace et de l'organisation des données en mémoire. En effet, dans le cas simple où les pixels sont rangés régulièrement par x croissant puis y croissant en mémoire le nombre de cycles de chargement correspond au chargement des lignes de cache à mettre à jour et leur nombre et longueur dépendent évidemment du sens de déplacement. La durée de mise à jour dépend des paramètres L, la latence du chargement d'une ligne, c'est-à-dire *grosso-modo* le temps d'accès au bus système, et de B, la bande passante, ou débit du bus, dans le meilleur cas (en mode burst). Nous avons donc, selon les cas suivants :

- Horizontal : $D_x = t_y \left(L + \frac{\delta_{t_x}}{B} \right)$
- Vertical : $D_y = \delta_{t_y} \left(L + \frac{t_x}{B} \right)$

Pour réduire la durée de mise à jour dans le cas d'un déplacement horizontal, qui est le plus pénalisant, nous avons à résoudre la contradiction entre :

- La réduction de la taille de la zone en cache
- L'augmentation relative de Δ_{t_x}, la zone mise à jour pour un déplacement horizontal, par rapport à la latence L.

Une autre solution est de modifier l'organisation des données en mémoire en répartissant le terme B, qui correspond au nombre de pixels consécutifs en x chargés par cycle d'horloge. Si chaque mot mémoire contient un rectangle de taille (B_x, B_y), nous avons :

- Horizontal : $D_x = \frac{t_y}{B_y} \left(L + \frac{\delta_{t_x}}{B_x} \right)$
- Vertical : $D_y = \frac{\delta_{t_y}}{B_y} \left(L + \frac{t_x}{B_x} \right)$

Dans ce modèle, la latence est potentiellement divisée par B_y, ou, du moins le temps d'accès au bus est partagé sur le chargement de plusieurs lignes dans le même burst.

3.2.2. Analyse du cache simple

L'efficacité du cache, mesurée par l'inverse du nombre de cycles d'attente, est liée à son comportement et à sa stabilité et dépend de ses fréquences de coupure et vitesse de déplacement. Nous cherchons à le régler de manière à permettre un compromis intéressant entre :

- La taille des mémoires internes du cache, qui doit être minimisée
- Le recouvrement des accès au cache et sa mise à jour, qui doit être maximisé
- Le taux de défaut de cache, qui doit être minimisé.

En effet, la méthode d'analyse par filtrage des coordonnées induit nécessairement un retard de phase qui doit être compensé par une estimation de la vitesse de déplacement du cache. Dans

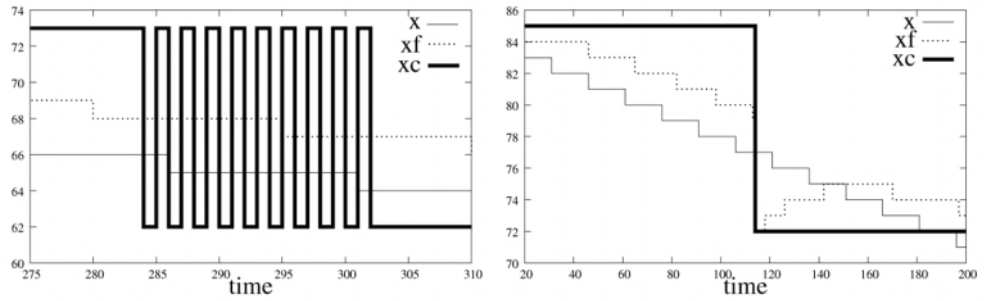


Figure 4. Poursuite a vitesse $\alpha > 2\gamma$ sans et avec stabilisation. En gras la position du centre du cache.

sa version actuelle, la vitesse est proportionnelle à la variance estimée et le concepteur doit déterminer ce paramètre α. Les fréquences de coupures a et b correspondent au nombre de points sur lesquels sont fait le calcul de la moyenne. Les contraintes imposées par le calcul en virgule fixe limitent la fréquence de coupure la plus basse, donc le nombre de points pris en compte, mais cela est compensé par un facteur de sous-échantillonnage qui permet d'abaisser encore la fréquence de coupure lorsque les signaux sont très basse fréquence.

Les résultats de la section précédente nous indiquent que pour compenser la latence de la mémoire, α doit être assez grand et les fréquences de coupures assez basses. Cela permet d'obtenir des déplacements du cache importants et ainsi de réduire le coût du chargement des données, particulièrement dans le cas où l'on se déplace horizontalement.

Le paramètre α a une grande influence sur la stabilité de l'algorithme et nécessite l'adjonction d'un mécanisme de stabilisation. En effet, si α est supérieur à 2γ, le processus de suivi devient instable car lors d'un déplacement, le point P_f est dépassé et n'est pas dans la nouvelle zone de garde, ce qui produit un nouveau déplacement en sens inverse. Pour obtenir de grandes vitesses de déplacement et un suivi stable, le point P_f est forcé à la valeur du nouveau centre du cache, ainsi qu'illustré figure 4, et une contrainte de continuité limite le déplacement. Stabilisé, l'algorithme de prédiction est efficace et permet de réduire le nombre de déplacements (*cf.* figure 5).

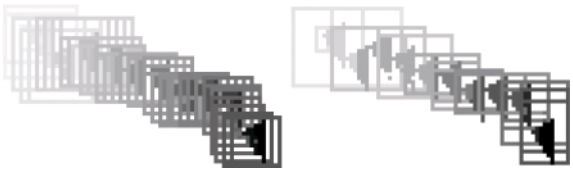


Figure 5. Poursuite pour $\alpha = 1$ et $\alpha = 3$ (les niveaux de gris représentent le temps). Les triangles sont les points utilisés par l'algorithme (un snake dans ce cas), les rectangles les bords des zones en cache.

Le recouvrement des accès au cache et sa mise à jour correspond au nombre de cycles pendant lesquels on peut faire des accès au cache avant qu'un défaut ne se produise et bloque les accès au cache[1]. Le recouvrement peut être amélioré en accroissant l'écart entre la zone de garde et la taille du cache car on peut estimer que la durée d'accès dans le cache est proportionnel à $t - g$, sur chaque direction. Nous avons donc intérêt à réduire g, tout en accroissant α, mais ceci rend le cache instable.

Le réglage des paramètres du cache est délicat et nécessite pour l'instant, en attendant de l'automatiser, une bonne compréhension des différents critères d'efficacité pour trouver une solution adaptée à l'application. À ce jour le réglage du cache est fait à partir d'une séquence d'accès supposée difficile à suivre (variations élevées puis lentes) et d'un modèle de bus (latence et débit) puis validée expérimentalement sur une plateforme matérielle, car des simulations seraient trop longues, par la mesure des temps de calcul sur toutes les séquences. Nous verrons que les résultats obtenus sont conformes aux modèles de simulation.

3.2.3. Cache hiérarchique

Le réglage d'un cache hiérarchique à un ou plusieurs niveaux nécessite des réglages différents selon le niveau dans la hiérarchie. En effet, les niveaux supérieurs analysent les requêtes des caches « inférieurs » qui correspondent à des chargements de zones de mise à jour alors que les caches « feuilles » analysent les requêtes de l'application.

Les caches du niveau n ont accès au cache de niveau $n + 1$ sur un bus à haut débit et latence réduite ce qui permet son partage avec un très bon taux de parallélisme. Le cache de niveau n est donc réglé en premier, en supposant un cache de niveau supérieur idéal puis le cache de niveau $n + 1$ est réglé à son tour, et ainsi de suite jusqu'au cache racine. Les caches feuilles se par-

1. En effet, l'arrêt du transfert d'une ligne depuis la mémoire principale est coûteux et il semble préférable de ne pas l'interrompre pour faire un défaut de cache. Les expériences faites sur l'application de rétro-projection nous indiquent, dans ce cas particulier, qu'il n'est pas non plus intéressant d'interrompre une mise à jour complète pour faire un défaut.

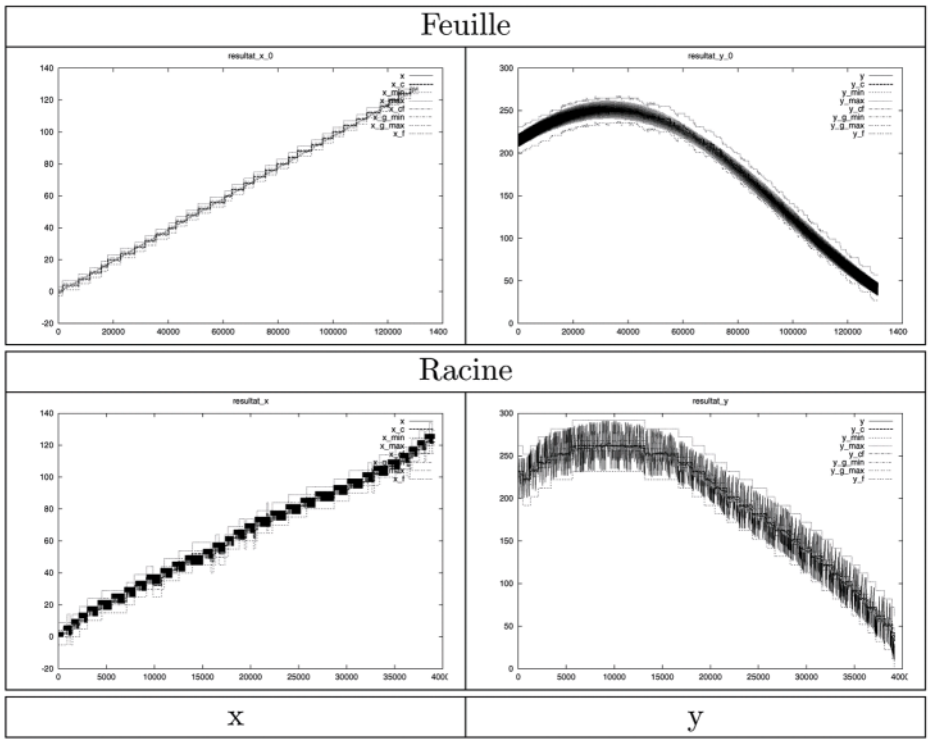


Figure 6. Séquences d'accès dans un cache hiérarchique.

tagent le **Bus Image**, et leur nombre influe sur le temps moyen d'accès au bus, celui-ci étant de latence L quasi nulle (environ 1 ou 2 cycles) une fois le bus acquis.

La figure 6 illustre ce phénomène pour un cache de 2 niveaux, avec 4 feuilles, chacune d'elle attachée à une unité de rétro-projection. Nous remarquons que chaque feuille fournie 131072 données à son unité de calcul mais le cache de niveau supérieur fourni 40000 données aux 4 feuilles. Le taux de réutilisation d'une donnée en cache feuille est de $\frac{131072}{10000} = 13$, ce qui est proche du taux théorique pire cas de $\frac{16}{\sqrt{2}} = 11{,}3$ et meilleur cas de 16. À noter que, avec ce réglage du cache, les 4*131072 accès mémoire (4 en parallèle) se font en 183429 cycles d'horloge, soit une perte d'environ 50000 cycles d'horloge qui inclus les temps d'initialisation et les défauts de cache (estimé à 2 %). Ces performances peuvent être améliorés pour obtenir un taux de défaut de cache de 0 %, tout simplement en augmentant la taille des feuilles mais au prix d'un accroîssement de la quantité de mémoire interne. En résumé, le cache hiérarchique permet bien de réduire le débit sur le bus principal.

3.3. Architecture système

Une plate-forme SOPC (System on Programmable Chip) a l'avantage de permettre une implémentation efficace et économique de la rétro-projection 2D. Pour davantage de flexibilité, et à coût réduit, le sinogramme et l'image reconstruite sont stockés en SDRAM, ou DDR-SDRAM, et la hiérarchie mémoire indispensable au recouvrement des calculs et accès mémoires, à grande latence, est construite à l'aide des blocs mémoire intégrés dans le circuit SOPC.

L'architecture présentée figure 7(a) est évaluée sur une carte qui dispose d'une interface PCI avec une station hôte pour permettre l'échange de données avec le système. Cette carte dispose d'un circuit Virtex 2 Pro 2VP20, de 32 MO de SDRAM et 128 MO de DDR-SDRAM et d'un bridge PCI, réalisé sur un FPGA Spartan. Les données sont échangées entre le PC hôte et la carte grâce à un mécanisme de synchronisation.

L'architecture système est constituée :

- d'un processeur PPC en charge de la synchronisation des calculs et des communications
- d'une unité de rétro-projection avec son cache 2D-AP illustré en figure 7(b).
- de mémoire externe

Pour bénéficier au maximum de la cohérence spatiale 2D et temporelle des calculs, l'unité de rétro-projection reconstruit un bloc de forme quelconque de pixels voisins de f^*. Par souci de simplicité les blocs sont des carrés de pixels mais des hexagones pourraient être utilisés. Les données du sinogramme sont fournies à l'unité de rétro-projection par le cache 2D-AP qui se charge de leur transfert depuis la mémoire externe, le module étant maître sur le bus PLB.

Pour réduire les temps de calculs, l'unité de rétro-projection est parallélisée et une hiérarchie mémoire fournit les données aux modules : chaque module obtient ses données d'un cache 2D-AP

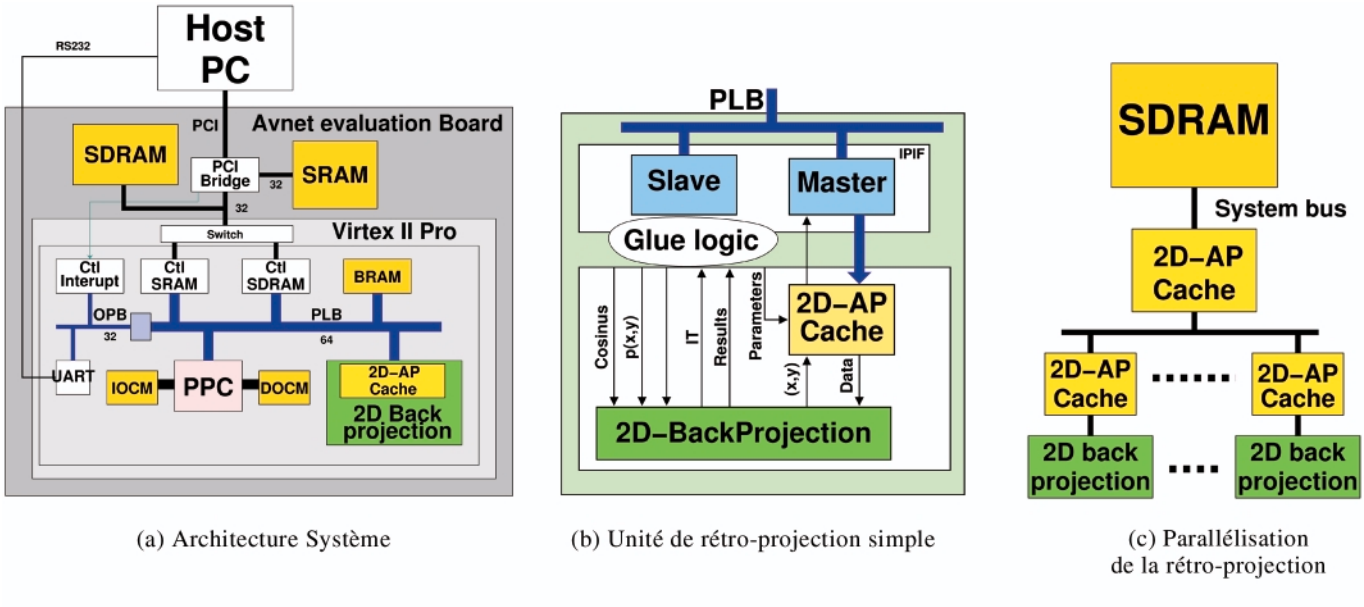


Figure 7. Architecture du système de rétro-projection.

qui lui même les met à jour à partir d'un cache 2D-AP de niveau supérieur. Dans notre exemple, nous nous arrêtons à un cache de niveau 2, comme illustré figure 7(c).
Cette architecture mémoire originale permet de n'utiliser qu'une seule mémoire externe qui contient tout le sinogramme, contrairement à la solution proposée par [3], tout en autorisant :

- le recouvrement de la latence de la mémoire externe et du bus système
- la réduction du débit à la mémoire externe.

Les performances de cette hiérarchie mémoire sont améliorées lorsque la latence se réduit et une connexion directe de la mémoire à l'unité de rétro-projection permettrait de paralléliser massivement les unités de rétro-projection.

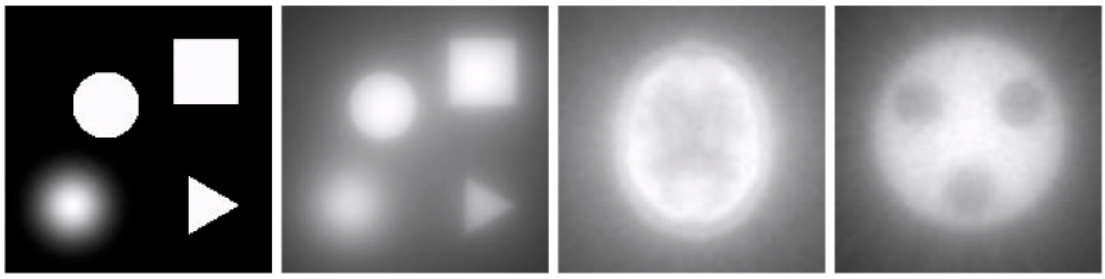
4. Résultats

Les objectifs annoncés de cette implémentations de la rétro-projection sont l'accélération des calculs à qualité de reconstruction égale. Cette dernière est une quantité difficile à mesurer, car elle dépend effectivement de l'objectif final. Par exemple, en clinique, l'habitude et la subjectivité des praticiens les conduit parfois à préférer des images classiques issues des algorithmes de rétro-projection plutôt que celles produites par des algorithmes itératifs plus « performants ».
Dans un premier temps, nous nous intéressons à l'erreur commise lors de la reconstruction qui peut être considérée comme la somme de l'erreur de méthode, due à l'algorithme de rétro-projection, et de l'erreur d'implémentation, due aux approximations liées au matériel et logiciel informatique.
Nous montrons que l'erreur d'implémentation due au passage en virgule fixe est très faible devant l'erreur de méthode.
Dans un deuxième temps, nous quantifions les gains de performances, tant en simulation qu'à partir de métriques mesurées sur la plateforme par rapport à une machine séquentielle classique de même génération technologique.

4.1. Qualité de la reconstruction

4.1.1. Protocole de validation

Nous montrons que l'erreur de méthode est très largement prépondérante sur l'erreur d'implémentation dès que le nombre de bits significatifs utilisés pour les calculs en virgule fixe dépasse une certaine valeur (9 bits), ce qui justifie le terme de qualité de reconstruction égale.
Pour évaluer l'erreur de méthode, nous avons construit des sinogrammes à partir d'une image définie analytiquement et contenant des formes simples (carré, cercle). Cette image est donc une référence exacte de même que les sinogrammes calculés et permet d'évaluer l'erreur de méthode.
Pour évaluer l'erreur d'implémentation, nous allons comparer les résultats de reconstruction obtenus par différentes implémentations : calcul en utilisant des réels double précision en virgule flottante, calcul en utilisant une notation en virgule fixe avec un nombre paramétré de décimales. Par ailleurs, le calcul de la transformée de radon nous donne des coordonnées de points image en valeur réelle. Les interpolations testées sont soit la troncature, soit l'interpolation linéaire. Les images utilisées pour cette évaluation sont des sinogrammes simulés d'images cérébrales et d'un fantôme cylindrique. Les images d'origine ne sont pas disponibles et seule une inter-comparaison peut avoir lieue.



(a) Image originale (analytique) (b) Image reconstruite (c) Donnée simulée (d) Donnée simulée

Figure 8. Images reconstruites à partir de données analytiques ou simulées.

Tableau 1. Erreurs de méthode pour différentes implémentations.

	Erreur quadratique moyenne : eqm $\cdot 10^{-5}$		
	réel interpolé	réel	virgule fixe (2+15 bits)
Dessins (analytique)	301.65	301.65	301.65
Fantôme (analytique)	282.22	282.22	282.22

Pour quantifier la qualité de reconstruction, nous calculons l'erreur quadratique moyenne de l'image I par rapport à l'image I_{ref} de la manière suivante :

$$\mathrm{eqm}(I/I_{ref}) = \frac{\sqrt{\sum_{x=0}^{x_{max}} \sum_{y=0}^{y_{max}} \left(\frac{I(x,y)-I_{ref}(x,y)}{I(x,y)+I_{ref}(x,y)} \right)^2}}{x_{max} * y_{max}} \qquad (3)$$

La base de donnée est constituée de :

- sinogrammes calculées de manière analytique par transformation de Radon à partir de formes géométriques simples. [20]
- données générées par le simulateur monte carlo PET-SORTEO d'Anthonin Reilhac. [5]

Pour les données analytiques, l'image originale existe. Nous sommes donc en mesure de calculer l'erreur de méthode : $\mathrm{eqm}(I_{reconstruit}/I_{original})$. Nous calculons également pour l'ensemble des données l'erreur quadratique moyenne entre différentes implémentations de la rétro-projection et celle la plus exacte utilisant des réels double précision avec interpolation linéaire.

4.1.2. Erreur de méthode

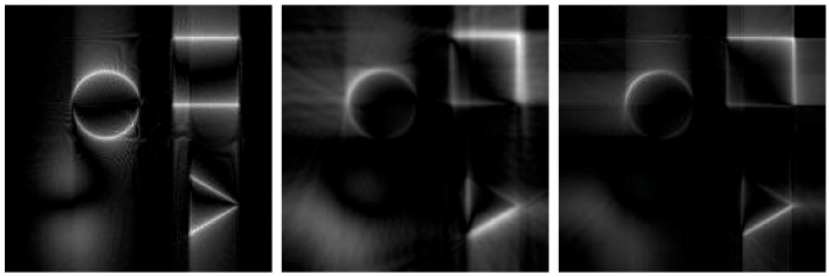
Le Tableau 1 présente l'erreur quadratique moyenne de l'image reconstruite à partir de sinogrammes calculés par transformation de Radon. Cette erreur est calculée par rapport à l'image d'origine et pour les différentes implémentations de la rétro-projection: réel, réel avec interpolation linéaire et virgule fixe avec $2+15=17$ bits de précision (2 pour la partie entière et 15 pour la partie décimale). Elle caractérise l'erreur de méthode inhérente à la rétro-projection qui n'est pas l'exacte transformée inverse de la projection. En effet, la rétro-projection laissent des résidus répartis en étoile autour des points d'activité (voir figure 8). L'erreur de méthode est importante de l'ordre de 3 % pour les différentes implémentation.

4.1.3. Écarts relatifs entre implémentations

Les écarts entre images reconstruites pour différentes implémentations de l'algorithme et celle en réel interpolé sont présentés dans le tableau 2 et sur la figure 10.

Erreur entre les méthodes interpolées et non interpolées
Nous observons que l'image reconstruite sans interpolation (figure 9(a)) est légèrement décalée vers le bas, ce qui s'explique par l'origine des erreurs de calcul. Pour simplifier la démonstration, nous faisons les calculs pour la reconstruction des pixels $(x,0)$ et $(0,y)$.

Pour reconstruire le pixel $f_t^*(x,0)$, nous calculons $f_t^*(x,0) = \sum_{k=0}^{K} S(k, r_k(x,0)), r_k(x,0) = E[x\cos(\theta_k) + \mathrm{offset}]$, ou encore $r_k(x,0) = x\cos(\theta_k) + \delta r + \mathrm{offset}$ avec $\delta r \in [0,1[$, et de moyenne $\frac{1}{2}$. Nous pouvons écrire $\delta r = a\sin(\theta_k)$. La moyenne de a est non nulle car $\sin(\theta_k) \geqslant 0$. Nous avons donc $f_t^*(x,0) = f^*(x,a)$, l'image reconstruite sans interpolation est donc l'image reconstruire idéale décalée de a pixels vers le bas. De la même façon, nous montrons qu'il n'y a pas de décalage horizontal car $f_t^*(0,y) = f^*(0,y)$, du fait de la symétrie du cosinus autour de $\frac{\pi}{2}$.



(a) Écart réel non interpolé (b) Écart virgule fixe (4 bits) (c) Écart virgule fixe (8 bits)

Figure 9. Écart entre implémentations par rapport à la méthode « réel interpolé » pour une image analytique.

Tableau 2. Écarts entre l'image reconstruite pour différentes implémentations de la rétro-projection avec celle en réel interpolé.

	Erreur quadratique moyenne : eqm $\cdot 10^{-5}$						
	réel	virgule fixe			virgule fixe interpolé		
		4 bits	8 bits	15 bits	4 bits	8 bits	15 bits
Dessins (analytique)	1,14	9,53	1,28	1,14	9,45	0,59	0,04
Fantôme (analytique)	0,81	8,48	0,94	0,81	8,42	0,5	0,03
Cerveau (simulé)	3,27	13,08	3,38	3,27	12,77	0,97	0,11
Cylindre et inserts (simulé)	4,09	16,10	4,23	4,09	15,64	1,13	0,13

Erreurs entre les calculs virgule flottante et virgule fixe
Les erreurs de reconstruction du calcul en virgule fixe s'accroissent lorsqu'on s'éloigne du centre de l'image (figures 10). En effet, les arrondis sur les valeurs des cosinus et sinus sont amplifiées lors du calcul de $x\cos(\theta_k) + y\sin(\theta_k)$. La précision nécessaire au codage des cosinus et sinus est tout simplement le nombre de bits nécessaires à la représentation des coordonnées x et y.

Discussion Le graphe de la figure 10 représente l'écart entre les images reconstruites en virgule fixe et virgule fixe interpolée avec celles reconstruites en réel interpolé en fonction du nombre de bits codant la partie décimale. À partir de 12 bits, les implémentations en entier offrent un résultat quasiment égal à l'implémentation référence en réel interpolé. L'écart relatif entre les différentes implémentations devient alors très faible, plus de 300 fois inférieur à l'erreur de méthode (*c.f.* tableau 1).
Ainsi l'implémentation en matériel qui est strictement équivalente à l'implémentation software en entier n'altère pas ou très peu la qualité de reconstruction.

4.2. Performances de l'architecture

Les mesures effectuées sur la plate-forme et les résultats de simulation nous montrent que la hiérarchie mémoire choisie est efficace et nous permet des accélérations importantes par la parallélisation des opérateurs et ceci en n'ayant qu'une mémoire externe qui contient toutes les données. Les métriques obtenues sont données dans le tableau 3 et montrent une accélération quasi-linéaire avec le nombre d'opérateurs. Elles correspondent à la reconstruction d'une image $x_{max} * y_{max} = 320 * 320$ à partir d'un sinogramme de résolution angulaire $K = 512$. L'image est reconstruite par blocs de taille $16 * 16$, qui est un

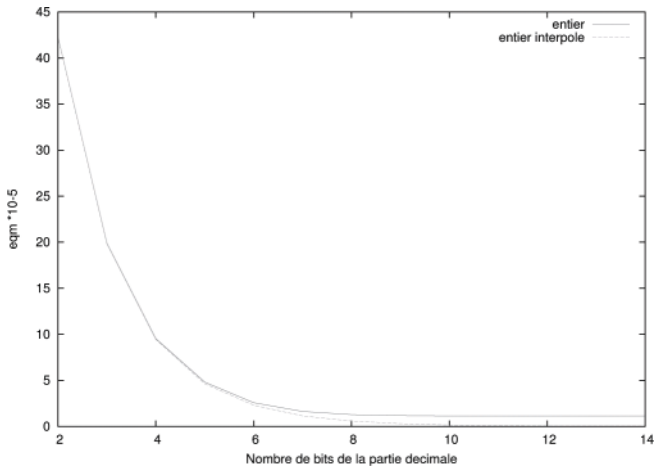


Figure 10. Erreur quadratique moyenne en fonction de la précision des méthodes en virgule fixe.

Tableau 3. Performances de l'accélération de la rétro-projection.

Système	Cycles	Temps @50 MHz	Temps @175 MHz
Logiciel			
Pentium 3 (1 GHz)		3,5 s	
PPC (VirtexII-Pro)		94 s	26 s
Matériel simple			
Idéal	52.10^6 $(320 * 320 * 512)$	1,1 s	0,3 s
Sans cache	$52.10^6 * 28$ (Idéal∗Latence)	30 s	8,5 s
1 unité	78.10^6	1,5 s	0,42 s
	Matériel parallélisé		
2 unités	42.10^6	0,8 s	0,22 s
4 unités	21.10^6	0,42 s	0,12 s
9 unités	11.10^6 (simulé)	0,22 s	0,06 s

bon compromis entre la taille du cache, les ressources matérielles et l'efficacité de la reconstruction (*c.f.* section 2).

Idéalement, nous devrions pouvoir réaliser une reconstruction sans temps mort, c'est-à-dire en $x_{max} * y_{max} * K = 320 * 320 * 512$ cycles d'horloge. Ces performances idéales sont altérées du fait d'une part de la synchronisation entre l'opérateur matériel et le PPC et d'autre part des performances du Cache 2D-AP sur le bus système. Les mesures effectuées nous montrent que les simulations réalisées pour la reconstruction d'un bloc sont fiables et peuvent être extrapolées à la reconstruction d'une image complète (résultats de la reconstruction par 9 unités).

L'ensemble du module a été décrit en VHDL générique paramétrable pour explorer rapidement les différentes configurations possibles. Le cache est entièrement modulaire de façon à pouvoir mesurer l'efficacité de différents type d'estimateurs prédictifs et pour pouvoir construire une hiérarchie par simple assemblage de blocs. Le tableau 4 donne la complexité du système en nombre de LUT pour une synthèse sur cible Xilinx Virtex II-Pro, avec les paramètres du cache qui nous donnent les performances annoncées précédemment. Nous remarquons que l'accroissement de complexité du cache, entre 2 et 4 unités, est linéaire à une constante près, qui est la complexité du cache de niveau supérieur. Le placement routage sur un Virtex II-Pro nous permet d'atteindre 50 MHz et les tests de synthèse sur les Virtex 4-FX, la génération suivante, nous montrent qu'il est possible d'atteindre une fréquence d'horloge 175 MHz.

Tableau 4. Synthèse de l'IP et son cache 2D-AP sur cible Virtex II-Pro.

Module	CLB	FG
1 unité	1078	2155
2 unités	2253	4506
4 unités	3877	7753

5. Conclusion

Nous avons présenté dans cet article un système de reconstruction par rétro-projection 2D implémenté et validé sur une plateforme SOPC.

Ce système a été conçu d'une part pour résoudre le goulot d'étranglement existant lors des accès en mémoire et de manière à réduire les erreurs de reconstruction dues aux calculs en virgule fixe. Lors de la reconstruction, les données nécessaires sont prédites statistiquement afin que le cache puisse les pré-charger avant que l'unité de rétro-projection ne les utilise effectivement. De plus, l'algorithme de rétro-projection a été implémenté de façon à augmenter la localité spatiale 2D et temporelle, l'utilisation du cache en devient plus pertinente.

La rétro-projection est utilisée par les algorithmes itératifs plus sophistiqués. Ces derniers offrent une meilleure qualité d'image mais en contrepartie ont un temps reconstruction beaucoup plus long. La réalisation présentée dans cet article est une première étape pour construire un système de reconstruction itératif, adaptable et flexible. Implémenté sur un SOPC, le module de rétro-projection 2D peut être dupliqué pour paralléliser les calculs. Un même module de cache peut être partagé ou une hiérarchie de caches mémoire peut être mise en place. Ainsi en s'appuyant sur la localité temporelle et spatiale 2D, nous pouvons reconstruire simultanément plusieurs pixels et la vitesse de reconstruction est alors notablement améliorée.

Références

[1] S. BASU, Y. BRESLER, O (n2logn) backprojection algorithm for tomography. *IEEE Trans. Med. Imaging*, 19(8):1760-1773, 2000.

[2] D. BRASSE, B. HUMBERT, C. MATHELIN, M.C. RIO, J.L. GUYONNET, Towards an inline reconstruction architecture for micro-ct systems. *Phys. Med. Biol.*, 50:5799-5811, 2005.

[3] S. CORIC, M. LEESER, E. MILLER, M. TREPANIER, Parallel-beam backprojection: an FPGA implementation optimized for medical imaging. *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, 39(3), 2005.

[4] S. CORIC, M. LEESER, E. MILLER, M. TREPANIER, Parallel-beam backprojection: an fpga implementation optimized for medical imaging. In *FPGA '02: Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, pages 217-226, New York, NY, USA, 2002. ACM Press.

[5] A. REILHAC *et al*, Pet-sorteo: validation and development of database of simulated pet volumes. *IEEE Tr. Nucl. Sci.*, 52(5):1321-1328, October 2005.

[6] P. GRANGEAT, *La Tomographie medicale*. Hermes, 2002.

[7] D. KIM, R. MANAGULI, Y. KIM, Data cache and direct memory access in programming mediaprocessors. *IEEE Micro*, 21(4):33-42, 2001.

[8] J.F. LECOMTE, *Quantification d'images tomographiques 3D et détection appliquées à l'imagerie cérébrale fonctionnelle individuelle*. PhD thesis, Universite de Caen, 1997.

[9] M. W. LENOX, T. GREMILLION, S. MILLER, J. W. YOUNG, Coincidence time alignment for planar pixellated positron emission tomography detector arrays. In *IEEE Nuclear Science Symposium*, pages 1952-1954. IEEE Computer Society, 2001.

[10] X. LI, J. NI, G. WANG, Parallel iterative cone beam ct image reconstruction on a pc cluster. *Journal of X-Ray SCiences and technology*, 13:1-10, 2005.

[11] B. DE MAN, S. BASU, Distance-driven projection and backprojection in three dimensions. *Phys. Med. Biol*, 49:2463-2475, 2004.

[12] A. MEHTA, J. ROTEM, Fpga co-processing solutions for signal processing applications. *ece*, pages 25-26, 2005.

[13] K. MUELLER, R. YAGEL, Rapid 3d cone-beam reconstruction with sart using texture mapping hardware. *IEEE Trans. Med. Imag*, 19(12):1227-1237, 2000.

[14] D.M. PETTERSEN, S. MIKKELSEN, J. TALEBI, D. MEIER, A readout asic for spect. *IEEE Trans. Nucl. Sci*, 52(3):764--771, 2005.

[15] T. PIPATSRISAWAT, A. GACIC, F. FRANCHETTI, M. PÜSCHEL, J. M. MOURA, Performance analysis of the filtered backprojection image reconstruction algorithms. In *ICASSP 2005*, volume~5, pages 153-156, 2005.

[16] K. RAJAN, L. M. PATNAIK, Cbp and art image reconstruction algorithms on media and dsp processors. *Microprocessors and Microsystems*, 25(5):233-238, 2001.

[17] T. RODET, L. DESBAT, P. GRANGEAT, Parallel algorithm based on a frequential decomposition for dynamic 3d computed tomography. In *Parallel and Distributed Processing Symposium*, pages 14-21. IEEE Computer Society, 2003.

[18] P. SACK, A.C. ELSTER, Fast mpi broadcasts through reliable multicasting. In *6th International Conference on Applied Parallel Computing Advanced Scientific Computing*, volume 2367 of *Lecture Notes In Computer Science*, pages 445-453. Springer Verlag, London, UK, June 2002.

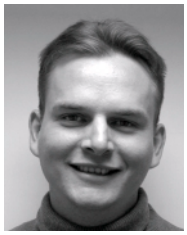
[19] L. G. STRAUSS, A. DIMITRAKOPOULOU-STRAUSS, G. KONTAXAKIS, Performance characteristics of iterative image reconstruction techniques for routine use in positron emission tomography. *Alasbimn Journal*, 2001.
http://www.alasbimnjournal.cl/revistas/13/strausstotal.html.

[20] P. TOFT., *The Radon Transform : Theory and Implementation*. PhD thesis, Technical University of Denmark, Copenhagen, 1996.

[21] S. VOLLMAR, C. MICHEL, J. T. TREFFERT, D. F. NEWPORT, M. CASEY, C. KNOSS, K. WIENHARD, X. LIU, M. DEFRISE, W.-D. HEISS, HeinzelCluster: accelerated reconstruction for FORE and OSEM3D. *IOP Phys. Med. Biol*, 47:2651-2658, 2002.

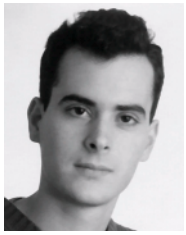
[22] F. XU, K. MUELLER, Accelerating popular tomographic reconstruction algorithms on commodity pc graphics hardware. *IEEE Trans. Nucl. Sci*, 52(3):654-664, 2005.

[23] J.W. YOUNG, J.C. MOYERS, M. LENOX, Fpga based front-end electronics for a high resolution PET scanner. *IEEE Trans. Nuclear Science*, 47:1676-80, 2000.



Stéphane **Mancini**

Stéphane Mancini est docteur (2000) et ingénieur (1996) de Telecom Paris. Il est actuellement maître de conférence à l'ENSERG/INPG en conception des SoC et effectue ses recherches au LIS sur la thématique de l'Adéquation Algorithme Architecture. Entre autre, il est responsable du projet ArchiTEP, qui vise à accélérer la reconstruction TEP 2D et 3D.



Nicolas **Gac**

Nicolas Gac est actuellement doctorant au LIS. Titulaire d'un diplôme d'ingénieur au département TELECOM de l'INP Grenoble et d'un DEA en SIPT, il effectue depuis Octobre 2004 une thèse sur l'adéquation algorithme architecture pour la reconstruction 3D en imagerie médicale TEP sous la direction de Michel Desvignes et Stéphane Mancini.

Michel **Desvignes**

Michel Desvignes a obtenu son diplôme d'ingénieur à l'ENSI-CAEN en 1985. Il a soutenu sa thèse en 1990 puis son habilitation à diriger les recherches en 2002 à l'Université de Caen. Il est actuellement professeur à l'ENSERG/INPG en informatique. Ses activités de recherche concernent le traitement d'images et la reconnaissance de formes, particulièrement dans les domaines de l'imagerie médicale et de la vidéo.

Olivier **Bourrion**

Olivier Bourrion, ingénieur en Génie électrique option micro-électronique de l'INSA Lyon, est ingénieur de recherche au Laboratoire de Physique Subatomique et de Cosmologie de Grenoble. Son domaine d'activité principal est l'acquisition de donnée, ce qui implique fréquemment la mise en œuvre de composants programmables et de micro-contrôleurs.

Olivier **Rossetto**

Olivier Rossetto a obtenu en 1991 un doctorat en micro-électronique de l'INP Grenoble. Depuis 1993, il est enseignant chercheur à l'université Joseph Fourier de Grenoble, ou ses domaines d'intérêt portent sur l'intégration VLSI de systèmes d'instrumentation et de traitement pour la physique des particules et les systèmes d'imagerie médicale TEP.