Université de Montréal

Les algorithmes de haute résolution en tomographie d'émission par positrons : développement et accélération sur les cartes graphiques

par
Moulay Ali Nassiri

Département de physique
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en physique

Mai, 2015

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée:

Les algorithmes de haute résolution en tomographie d'émission par positrons : développement et accélération sur les cartes graphiques

présentée par:

Moulay Ali Nassiri

a été évaluée par un jury composé des personnes suivantes:

Francois Schiettekatte,	président-rapporteur
Claude Leroy,	directeur de recherche
Jean-François Carrier,	codirecteur
Philippe Després,	codirecteur
Rickard Blunk,	membre du jury
Roger Lecomte,	examinateur externe
Patrice Marcotte,	représentant du doyen de la FES

Thèse acceptée le: 1er mai 2015

RÉSUMÉ

La tomographie d'émission par positrons (TEP) est une modalité d'imagerie moléculaire utilisant des radiotraceurs marqués par des isotopes émetteurs de positrons permettant de quantifier et de sonder des processus biologiques et physiologiques. Cette modalité est surtout utilisée actuellement en oncologie, mais elle est aussi utilisée de plus en plus en cardiologie, en neurologie et en pharmacologie. En fait, c'est une modalité qui est intrinsèquement capable d'offrir avec une meilleure sensibilité des informations fonctionnelles sur le métabolisme cellulaire. Les limites de cette modalité sont surtout la faible résolution spatiale et le manque d'exactitude de la quantification. Par ailleurs, afin de dépasser ces limites qui constituent un obstacle pour élargir le champ des applications cliniques de la TEP, les nouveaux systèmes d'acquisition sont équipés d'un grand nombre de petits détecteurs ayant des meilleures performances de détection. La reconstruction de l'image se fait en utilisant les algorithmes stochastiques itératifs mieux adaptés aux acquisitions à faibles statistiques. De ce fait, le temps de reconstruction est devenu trop long pour une utilisation en milieu clinique. Ainsi, pour réduire ce temps, on les données d'acquisition sont compressées et des versions accélérées d'algorithmes stochastiques itératifs qui sont généralement moins exactes sont utilisées. Les performances améliorées par l'augmentation de nombre des détecteurs sont donc limitées par les contraintes de temps de calcul.

Afin de sortir de cette boucle et permettre l'utilisation des algorithmes de reconstruction robustes, de nombreux travaux ont été effectués pour accélérer ces algorithmes sur les dispositifs GPU (*Graphics Processing Units*) de calcul haute performance. Dans ce travail, nous avons rejoint cet effort de la communauté scientifique pour développer et introduire en clinique l'utilisation des algorithmes de reconstruction puissants qui améliorent la résolution spatiale et l'exactitude de la quantification en TEP.

Nous avons d'abord travaillé sur le développement des stratégies pour accélérer sur les dispositifs GPU la reconstruction des images TEP à partir des données d'acquisition en mode liste. En fait, le mode liste offre de nombreux avantages par rapport à la reconstruction à partir des sinogrammes, entre autres : il permet d'implanter facilement et avec précision la correction du mouvement et le temps de vol (TOF : *Time-Of Flight*) pour améliorer l'exactitude de la quantification. Il permet aussi d'utiliser les fonctions de bases spatio-temporelles pour effectuer la reconstruction 4D afin d'estimer les paramètres cinétiques des métabolismes avec exactitude. Cependant, d'une part, l'utilisation de ce mode est très limitée en clinique, et d'autre part, il est

surtout utilisé pour estimer la valeur normalisée de captation SUV qui est une grandeur semi-quantitative limitant le caractère fonctionnel de la TEP. Nos contributions sont les suivantes :

- Le développement d'une nouvelle stratégie visant à accélérer sur les dispositifs GPU l'algorithme 3D LM-OSEM (*List Mode Ordered-Subset Expectation-Maximization*), y compris le calcul de la matrice de sensibilité intégrant les facteurs d'atténuation du patient et les coefficients de normalisation des détecteurs. Le temps de calcul obtenu est non seulement compatible avec une utilisation clinique des algorithmes 3D LM-OSEM, mais il permet également d'envisager des reconstructions rapides pour les applications TEP avancées telles que les études dynamiques en temps réel et des reconstructions d'images paramétriques à partir des données d'acquisitions directement.
- Le développement et l'implantation sur GPU de l'approche Multigrilles/Multitrames pour accélérer l'algorithme LMEM (*List-Mode Expectation-Maximization*). L'objectif est de développer une nouvelle stratégie pour accélérer l'algorithme de référence LMEM qui est un algorithme convergent et puissant, mais qui a l'inconvénient de converger très lentement. Les résultats obtenus permettent d'entrevoir des reconstructions en temps quasi-réel que ce soit pour les examens utilisant un grand nombre de données d'acquisition aussi bien que pour les acquisitions dynamiques synchronisées.

Par ailleurs, en clinique, la quantification est souvent faite à partir de données d'acquisition en sinogrammes généralement compressés. Mais des travaux antérieurs ont montré que cette approche pour accélérer la reconstruction diminue l'exactitude de la quantification et dégrade la résolution spatiale. Pour cette raison, nous avons parallélisé et implémenté sur GPU l'algorithme AW-LOR-OSEM (*Attenuation-Weighted Line-of-Response-OSEM*) ; une version de l'algorithme 3D OSEM qui effectue la reconstruction à partir de sinogrammes sans compression de données en intégrant les corrections de l'atténuation et de la normalisation dans les matrices de sensibilité. Nous avons comparé deux approches d'implantation : dans la première, la matrice système (MS) est calculée en temps réel au cours de la reconstruction, tandis que la seconde implantation utilise une MS pré-calculée avec une meilleure exactitude. Les résultats montrent que la première implantation offre une efficacité de calcul environ deux fois meilleure que celle obtenue dans la deuxième implantation. Les temps de reconstruction rapportés sont compatibles avec une utilisation clinique de ces deux stratégies.

Keywords : Tomographie d'émission par positrons, TEP, reconstruction, algorithmes, haute résolution, accélération, GPU, mode liste, sinogrammes, matrice de sensibilité.

ABSTRACT

Positron emission tomography (PET) is a molecular imaging modality that uses radiotracers labeled with positron emitting isotopes in order to quantify many biological processes. The clinical applications of this modality are largely in oncology, but it has a potential to be a reference exam for many diseases in cardiology, neurology and pharmacology. In fact, it is intrinsically able to offer the functional information of cellular metabolism with a good sensitivity. The principal limitations of this modality are the limited spatial resolution and the limited accuracy of the quantification. To overcome these limits, the recent PET systems use a huge number of small detectors with better performances. The image reconstruction is also done using accurate algorithms such as the iterative stochastic algorithms. But as a consequence, the time of reconstruction becomes too long for a clinical use. So the acquired data are compressed and the accelerated versions of iterative stochastic algorithms which generally are non convergent are used to perform the reconstruction. Consequently, the obtained performance is compromised.

In order to be able to use the complex reconstruction algorithms in clinical applications for the new PET systems, many previous studies were aiming to accelerate these algorithms on GPU devices. Therefore, in this thesis, we joined the effort of researchers for developing and introducing for routine clinical use the accurate reconstruction algorithms that improve the spatial resolution and the accuracy of quantification for PET.

Therefore, we first worked to develop the new strategies for accelerating on GPU devices the reconstruction from list mode acquisition. In fact, this mode offers many advantages over the histogram-mode, such as motion correction, the possibility of using time-of-flight (TOF) information to improve the quantification accuracy, the possibility of using temporal basis functions to perform 4D reconstruction and extract kinetic parameters with better accuracy directly from the acquired data. But, one of the main obstacles that limits the use of list-mode reconstruction approach for routine clinical use is the relatively long reconstruction time. To overcome this obstacle we :

– developed a new strategy to accelerate on GPU devices fully 3D list mode ordered-subset expectation-maximization (LM-OSEM) algorithm, including the calculation of the sensitivity matrix that accounts for the patient-specific attenuation and normalisation corrections. The reported reconstruction are not only compatible with a clinical use of 3D LM-OSEM algorithms, but also lets us envision fast reconstructions for advanced PET

applications such as real time dynamic studies and parametric image reconstructions.

– developed and implemented on GPU a multigrid/multiframe approach of an expectation-maximization algorithm for list-mode acquisitions (MGMF-LMEM). The objective is to develop new strategies to accelerate the reconstruction of gold standard LMEM (list-mode expectation-maximization) algorithm which converges slowly. The GPU-based MGMF-LMEM algorithm processed data at a rate close to one million of events per second per iteration, and permits to perform near real-time reconstructions for large acquisitions or low-count acquisitions such as gated studies.

Moreover, for clinical use, the quantification is often done from acquired data organized in sinograms. This data is generally compressed in order to accelerate reconstruction. But previous works have shown that this approach to accelerate the reconstruction decreases the accuracy of quantification and the spatial resolution. The ordered-subset expectation-maximization (OSEM) is the most used reconstruction algorithm from sinograms in clinic. Thus, we parallelized and implemented the attenuation-weighted line-of-response OSEM (AW-LOR-OSEM) algorithm which allows a PET image reconstruction from sinograms without any data compression and incorporates the attenuation and normalization corrections in the sensitivity matrices as weight factors. We compared two strategies of implementation: in the first, the system matrix (SM) is calculated on the fly during the reconstruction, while the second implementation uses a precalculated SM more accurately. The results show that the computational efficiency is about twice better for the implementation using calculated SM on-the-fly than the implementation using pre-calculated SM, but the reported reconstruction times are compatible with a clinical use for both strategies.

Keywords : Positron emission tomography, PET, GPU, reconstruction, acceleration, list mode, sinograms, sensitivity matrix, multigrid, multiframe, LM-OSEM, LMEM, MGMF-LMEM, LOR-OSEM, AW-LOR-OSEM.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES

1D	Une dimension
2D	Deux dimensions
3D	Trois dimensions
API	Application Programming Interface
CPU	Unité centrale de traitement (Central Processing Unit)
CRC	Contrast Recovery Coefficient
ART	Algebric Reconstruction Techniques
CUDA	Compute Unified Device Architecture
DRAM	Dynamic Random Access Memory
DVS	Décomposition en valeurs singulières
FDA	U.S. Food and Drug Administration
FDG	Fluoro-déoxyglucose
FBP	Rétroprojection filtrée (Filtred Backprojection)
FMA	Fused Multiply-Add
Go	Gigaoctets
GPGPU	General-Purpose Computing on Graphics Processing Units
GPU	Carte graphique (Graphical Processing Unit)
IRM	Imagerie par résonance magnétique
ko	kilooctets
LMF	List Mode Format
LM-EM	List-Mode Expectation-Maximization

LM-OSEM	List Mode Ordered-Subset Expectation-Maximization
LOR	Lignes de réponses (Lines-Of-Response)
MAP	Maximum a Posteriori
MV	Maximum de Vraisemblance
Mo	Mégaoctets
MGMF-LMEM	Multigrid Multiframe approach of List-Mode Expectation-Maximization
MS	Matrice système
mrd	Différence maximale entre les couronnes (maximum ring difference)
OSEM	Ordered-Subsets Expectation Maximization
PMT	Photomultiplicateur
PTX	Parallel Thread eXecution
TOF	Temps de vol (Time-of-Flight)
TDM	Tomodensitométrie
TEMP	Tomographie par émission monophotonique
TEP	Tomographie par émission de positrons
TEP-IRM	Système TEP combiné au système IRM
TEP-TDM	Système TEP combiné au système TDM
To	Téraoctets
RSB	Rapport signal-bruit
SC	Stopping Criterion
SIMD	Single Instruction Multiple Data
SM	Matrix system
SMP	Streaming Multiprocesseurs
SP	Streaming Processor
SUV	Standardized Uptake Value

À

Mes défunts parents

Ma femme Ilhame

Mon fils Yassine

Mes filles Basma et Salma

REMERCIEMENTS

J'exprime tout d'abord mes remerciements, mon estime et ma reconnaissance à mon directeur de recherche M. Claude Leroy pour avoir accepté de m'encadrer et de me permettre de réaliser cette thèse de doctorat tant souhaitée.

Ce travail de thèse n'aurait pas eu lieu si le destin avait décidé de ne pas mettre dans mon chemin mon codirecteur de recherche M. Jean-François Carrier. Je lui exprime donc ma gratitude pour ses conseils et ses encouragements bienveillants, ainsi que pour sa grande disponibilité et son soutien durant les moments d'incertitude. Je garde de vous un homme humble et toujours disponible pour aider les autres.

Mes sincères remerciements vont à mon codirecteur de recherche M. Philippe Després pour son encadrement, sa patience et ses précieux conseils tant sur le plan scientifique qu'humain. Je me souviendrai de vous comme un homme spontané, intransigeant sur la qualité et la rigueur, mais aussi un homme qui est sincère et sensible aux difficultés des autres.

Je n'oublie pas de remercier aussi M. Stefan Michalowski pour sa disponibilité à répondre rapidement à mes demandes informatiques, et M. Hugo Bouchard pour ses encouragements.

Ma reconnaissance et mes remerciements vont à ma femme Ilhame pour son appui moral indéfectible et son soutien financier. Je remercie aussi mon fils Yassine, mes filles Basma et Salma qui m'ont comblé de beaucoup d'amour me procurant l'énergie nécessaire pour terminer cette thèse. Excusez-moi, si ces dernières années, j'ai failli aux devoirs de papa en combinant les études et le travail.

CHAPITRE 1

INTRODUCTION

La tomographie d'émission par positrons (TEP) est une modalité d'imagerie médicale qui mesure la distribution spatio-temporelle d'une molécule marquée par un émetteur de positrons. C'est une modalité d'imagerie fonctionnelle qui permet de quantifier *in vivo* la captation de radiotraceur dans les organes et les tissus ciblés. Elle permet ainsi d'étudier *in vivo* le métabolisme cellulaire, la perfusion tissulaire et la densité de récepteurs d'un système de transmission neuronale. Ces études fonctionnelles au niveau cellulaire sont très utiles pour l'oncologie, les neurosciences et les troubles du métabolisme. Par exemple, elles permettent, en oncologie, de bien évaluer le degré de malignité des tumeurs à partir des aspects fondamentaux de la prolifération cellulaire tels que l'augmentation de la consommation du glucose, de la synthèse des protéines ou celle des acides nucléiques.

Les performances des systèmes TEP ont beaucoup été améliorées cette dernière décennie. En effet, l'utilisation du mode d'acquisition 3D, l'introduction des détecteurs plus performants et plus petits et la multiplication de leur nombre par un facteur de l'ordre de 10 ont amélioré la résolution et le rapport signal-bruit (RSB) des images reconstruites. Cependant, la taille des données d'acquisition a aussi énormément augmenté, et, par conséquent, le temps de reconstruction est devenu long pour une utilisation clinique de routine.

Pour adapter le temps de calcul aux besoins cliniques, les processus de reconstruction sont typiquement simplifiés via des approximations au détriment de l'exactitude de la quantification. D'autre part, l'analyse des examens TEP en clinique repose, généralement, sur une inspection visuelle des images 3D de la distribution spatiale du traceur quantifiée par des indices semi-quantitatifs tels que la valeur de captation standardisée (SUV : *Standardized Uptake Value*). Or, cette approche limite le caractère fonctionnel intrinsèque des données TEP. En effet, elle n'estime pas les paramètres physiologiques et cinétiques qui sont potentiellement pertinents en clinique, mais dont la reconstruction avec une meilleure exactitude est relativement longue par rapport aux contraintes d'utilisation clinique.

Malgré que les systèmes TEP offrent la possibilité d'enregistrer les données en mode histogramme dans des sinogrammes et en mode liste, la reconstruction des images TEP en clinique se fait, généralement, à partir des données en sinogrammes par l'algorithme itératif stochas-

tique OSEM (*Ordered-Subsets Expectation Maximisation*). En effet, le mode histogramme permet d'accélérer la reconstruction en utilisant deux approches. La première consiste à redistribuer les données d'acquisition 3D sur des sinogrammes 2D et la deuxième compresse ces données en combinant plusieurs sinogrammes voisins dans la direction axiale (*mashing data*). Mais il a été démontré que ces deux approches affectent l'exactitude de la quantification [120, 241]. Dans l'objectif de réduire aussi le temps de reconstruction, le calcul de la matrice système (MS) qui modélise le système d'acquisition se fait typiquement en temps réel durant la reconstruction par des méthodes analytiques simplifiées en ignorant des phénomènes physiques qui faussent la quantification, entre autres la portée du positron, la non-colinéarité des deux photons, la pénétration dans les cristaux et la diffusion inter-cristaux. De plus, l'algorithme OSEM, qui est l'algorithme le plus utilisé en clinique actuellement, est une version plus rapide, mais non convergente de l'algorithme MLEM (*Maximum-Likelihood Expectation-Maximisation*)[84, 178, 232]. Par contre, ce dernier est un algorithme robuste et convergent, mais il est très lent pour être utilisé en clinique de routine pour les acquisitions 3D [131].

Les paramètres physiologiques et cinétiques tels que le taux de perfusion, le taux de métabolisme sont aussi, généralement, estimés à partir des acquisitions dynamiques enregistrées en sinogrammes pour accélérer la reconstruction. La méthode la plus utilisée pour estimer ces paramètres consiste à construire les images dynamiques de la distribution spatiale du traceur et puis à calculer ces derniers par des fonctions analytiques à partir de ces images. Le problème avec cette approche est que les images dynamiques reconstruites présentent une mauvaise résolution temporelle et, par conséquent, les paramètres physiologiques et cinétiques estimés sont biaisés. Des auteurs ont amélioré le niveau du bruit de ces dernières en les estimant directement à partir des sinogrammes [233]. Mais même avec cette dernière approche, la limite de la résolution temporelle des acquisitions est un facteur qui a un grand impact sur l'exactitude des paramètres physiologiques. En effet, les acquisitions dynamiques sont un compromis entre la résolution temporelle et le RSB [186].

Par ailleurs, la reconstruction des données en mode liste a suscité un grand intérêt au cours de la dernière décennie. Ce mode offre de nombreux avantages sur le mode histogramme, entre autres : il permet d'implanter facilement et avec précision la correction du mouvement [105, 184] et le temps de vol (TOF : *Time-Of Flight*) pour améliorer l'exactitude de la quantification [45, 46, 65]. L'avantage le plus important est qu'il rend possible la reconstruction 4D en utilisant des fonctions de bases spatio-temporelles, ce qui permet d'estimer les paramètres cinétiques avec

exactitude directement à partir des données d'acquisition [186, 191, 192, 249].

Cependant, la reconstruction à partir des données en mode liste est relativement longue, surtout pour les grandes acquisitions dans lesquelles le nombre d'événements est plus grand que le nombre de lignes de réponses (LOR : *Line-Of-Response*) du système. De plus, le temps de calcul de la matrice de sensibilité pour les reconstructions en mode liste est aussi un obstacle à l'introduction de cette approche en clinique [127, 175]. En fait, cette matrice doit être calculée pour chaque patient à partir de l'image des coefficients d'atténuation obtenue par une acquisition tomodensitométrique. Pour les acquisitions dynamiques, cette matrice de sensibilité doit être calculée pour chaque fenêtre (*frame*) d'acquisition pour tenir compte du mouvement des organes.

Plusieurs travaux ont été effectués ces dernières années pour accélérer la reconstruction à partir du mode liste sur les cartes graphiques (GPU : *Graphical Processing Unit*) qui sont des plateformes non onéreuses de calcul parallèle. Mais le temps de calcul reste relativement long comparativement à la reconstruction à partir des sinogrammes. En effet, l'accès aux données enregistrées en mode liste dans la mémoire globale des GPU est difficilement optimisable. D'autre part, la majorité des travaux d'accélération a concerné surtout l'algorithme LM-OSEM (*List-Mode Ordered-Subset Expectation-Maximization*) lequel est un algorithme rapide, mais non convergent, ce qui peut compromettre l'avantage de reconstruction à partir du mode liste.

1.1 Les objectifs du projet

Les objectifs de ce projet de doctorat sont l'amélioration de la quantification en TEP et l'élargissement des applications cliniques de cette modalité, en introduisant en clinique de routine des algorithmes de reconstruction d'images plus rapides et plus robustes en terme de l'exactitude de la quantification. Ainsi, nous avons les objectifs de :

1. Accélérer la reconstruction de la matrice de sensibilité sur GPU pour le mode liste. L'objectif est de permettre l'utilisation du mode liste en clinique de routine, surtout pour les acquisitions dynamiques où la reconstruction à partir de ce mode est plus efficiente et plus exacte. Ce travail a fait l'objet d'une publication dans le journal *Physics in Medicine and Biology* [145] et d'une communication lors du congrès Fully 3D en 2011 [146].
2. Accélérer sur GPU l'algorithme LM-EM (*list-Mode Expectation-Maximization*), lequel est un algorithme convergent et puissant, mais qui a l'inconvénient de converger lentement. Pour pouvoir rendre l'exécution de cet algorithme quasi-temps réel, nous avons

utilisé la méthode qui consiste à commencer la reconstruction durant l'acquisition en utilisant un échantillonnage faible (grands voxels) et puis à améliorer l'échantillonnage au fur et à mesure que le nombre d'événements enregistrés augmente. Cette approche appelée MGMF-LMEM (*Multigrid Multiframe approach of List-Mode Expectation-Maximization*) a été inspirée du travail de Ranganath et al. [187] qui l'avait appliqué pour la reconstruction à partir des sinogrammes. Ce travail est accépté pour publication au journal Medical & Biological Engineering & Computing pour publication.

3. Étudier et explorer la possibilité d'accélérer sur GPU l'algorithme 3D AW-LOR-OSEM (*Attenuation-Weighted Line-of-Response-OSEM*) utilisant une MS précalculée. Cet algorithme est tout simplement une version de l'algorithme OSEM qui, à la différence de ce dernier, effectue la reconstruction à partir des sinogrammes non compressés en intégrant la correction de l'atténuation et de la normalisation dans les matrices de sensibilité pour améliorer l'exactitude de la quantification [96]. Puisque l'algorithme OSEM est la méthode de reconstruction la plus utilisée en clinique pour sa rapidité de convergence, alors l'objectif de ce travail est d'améliorer ses performances de quantification, mais sans détériorer son efficience de calcul par l'augmentation excessive de la taille des données de projection. L'utilisation d'une MS précalculée, au lieu d'une MS calculée en temps réel durant la reconstruction, permet de la déterminer avec plus d'exactitude sans augmenter beaucoup le temps de reconstruction [253]. Ce dernier travail a été présenté au congrès de l'AAPM 2011 à Vancouver [143] et au congrès Nuclear Science Symposium and Medical Imaging Conference 2011 à Valencia [142].

1.2 Organisation du document

Cette thèse est organisée en trois parties :

Partie 1 : Principes et algorithmes de reconstruction en TEP est composée de 3 chapitres. Le premier chapitre explique les principes physiques de base de la TEP, le deuxième chapitre présente les bases mathématiques et les méthodes de résolution du problème direct pour la reconstruction des images en TEP, et le troisième chapitre porte sur les différents algorithmes utilisés pour la résolution du problème inverse de la reconstruction des images.

Partie 2 : Matériels comprend les chapitres 5 et 6 qui présentent le matériel utilisé pour réaliser ce projet. Le chapitre 5 explique l'architecture physique des GPUs, les principes de base

de leur programmation et les méthodes d'optimisation des programmes GPUs. Le chapitre 6 présente brièvement le logiciel de modélisation Monte Carlo GATE et le logiciel STIR qui est un logiciel libre spécialisé dans la reconstruction des images TEP.

Partie 3 : Travaux et résultats constituée des chapitres 7, 8 et 9. Dans le chapitre 7, nous présentons notre travail portant sur l'accélération sur GPU de l'algorithme en mode liste LMEM intégrant le calcul de la matrice de sensibilité. Le chapitre 8 traite de l'algorithme MGMF-LMEM que nous avons développé et implanté sur GPU pour accélérer la reconstruction en TEP. Et le dernier chapitre présente notre travail de la mise en oeuvre sur GPU de l'algorithme haute résolution 3D LOR-OSEM.

Première partie

Principes et algorithmes de reconstruction en TEP

CHAPITRE 2

LES PRINCIPES PHYSIQUES DE LA TOMOGRAPHIE D'ÉMISSION PAR POSITRONS

2.1 Historique

La tomographie d'émission par positrons est le résultat d'une suite de découvertes dans le domaine de la physique, des mathématiques et de la biochimie durant le siècle dernier. Parmi ces découvertes, on cite : la formulation du principe mathématique de la reconstruction tomographique par Radon en 1917, l'annonce du postulat de l'existence des positrons par Dirac [53], la révélation expérimentale de ces positrons par Anderson [8], la découverte du phénomène de l'annihilation des positrons et l'émission de deux photons d'annihilation dans deux directions opposées par Thibaud [217], et les travaux de Warburg [236] sur le métabolisme du glucose par les cellules cancéreuses.

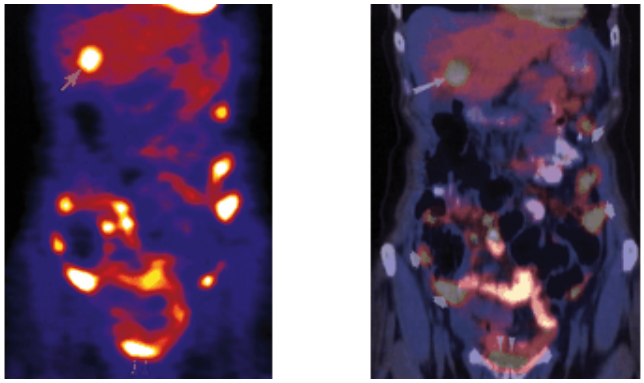
L'utilisation des isotopes émetteurs des positrons en imagerie médicale a été proposée la première fois par Wrenn et al. [243]. Le premier le système d'imagerie médicale par émission de positrons a été conçu par Brownell et Sweet [32]. C'était un appareil équipé avec deux détecteurs NaI (Tl) (iodure de sodium dopé au thallium) qui produisait des images planaires pour l'exploration du cerveau en utilisant les isotopes ^{64}Cu et ^{75}As. Les travaux de Cormack [43] et de Hounsfield [78] sur la tomographie axiale à partir des projections obtenues par la transmission des rayons X ont posé la base de la reconstruction tomographique en général. Le premier appareil TEP a été développé en 1975 par Ter-Pogossian et al. [216] sous le nom PETT II. Le système de détection, constitué d'un ensemble de 24 détecteurs NaI (Tl), est connecté à un système électronique de détection en coïncidence qui permet de réaliser la collimation. La méthode de reconstruction utilisée est la rétroprojection filtrée. Ce modèle a été amélioré et commercialisé en 1978 sous le nom de ECAT [167].

Par ailleurs, entre 1975 et 1985, l'utilisation du TEP était limitée à la recherche, et ce n'est qu'à partir de 1985 que cette modalité a été introduite en clinique. Pour les premières applications cliniques, le radiotraceur utilisé était le fluorodésoxyglucose (FDG) marqué au ^{18}F (^{18}FDG). Ce dernier est caractérisé par une demi-vie de 110 minutes, ce qui permet de le produire dans des centres de cyclotrons loin de l'installation TEP. Toutefois, l'implantation de la TEP est restée

limitée, au début, aux grands centres de médecine nucléaire équipés d'un cyclotron capable de produire les différents isotopes localement. C'est vers le début des années 90 que cette modalité a commencé à s'implanter rapidement en clinique, surtout en oncologie, grâce à l'approbation du traceur FDG par la FDA et à la mise en place d'un réseau de distribution de ce traceur. L'année 1998 a connu la fabrication du premier appareil combinant les deux modalités : la tomographie d'émission par positrons et la tomodensitométrie (TEP-TDM)[222]. La très grande majorité des systèmes TEP commercialisés actuellement sont des systèmes TEP-TDM.

2.2 Principe

La réalisation d'examens TEP nécessite l'injection d'un traceur marqué par un isotope émetteur de positrons. Ensuite, après un certain temps nécessaire pour que le radiotraceur soit capté, le patient passe dans le système TEP pour effectuer l'acquisition des données. En effet, le traceur est une molécule naturelle ou analogue qui participe aux processus biochimiques des organes ou des tissus ciblés. Puisque le marqueur est un émetteur de positrons, chaque émission (désintégration) sera accompagnée d'annihilation du positron avec un électron du voisinage et de l'émission simultanée de deux photons d'annihilation de 511 keV de directions opposées (figure 2.2). Une partie du nombre de ces paires de photons s'échappe du patient sans aucune interaction avec le milieu pour être détectée par le système de tomographie. Après l'acquisition des données, un système informatique estime la distribution spatio-temporelle du traceur dans le patient (figure 2.1). Ces images permettent, par la suite, de quantifier *in vivo* le métabolisme cellulaire, la perfusion des tissus et/ou la densité des récepteurs de la molécule injectée. Les isotopes les plus utilisés dans les examens TEP sont :^{18}F,^{11}C, ^{15}O et^{13}N. Ces éléments artificiels sont créés dans un cyclotron, qui est un accélérateur électromagnétique de haute fréquence. Le choix de ces isotopes est dicté par leurs propriétés chimiques, qui conditionnent leur affinité pour le marquage du traceur sans altérer ses propriétés métaboliques, et par leurs propriétés physiques telles que leur demi-vie et la portée du positron dans les tissus (tableau 2.I). En effet, les isotopes à très courte demi-vie nécessitent une infrastructure pour la production et le marquage des molécules sur place. La portée du positron est un facteur qui constitue une limite intrinsèque de la résolution spatiale de la TEP du fait que le lieu de l'émission du positron est différent du lieu de l'annihilation. La résolution se détériore avec l'augmentation de la portée du positron dans les tissus.



a) b)

Figure 2.1 – a) Image TEP obtenue en utilisant le traceur FDG, b) la même image TEP fusionnée avec l'image TDM.Tirées de Kapoor et al. [99].

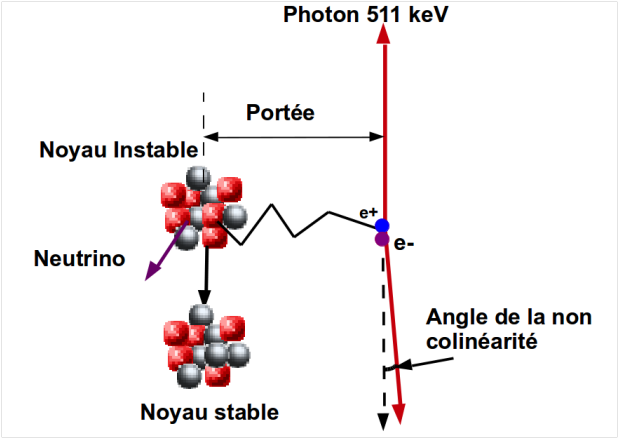


Figure 2.2 – Schéma du principe d'émission d'un positron et son annihilation avec un électron.

2.3 Émission du positron

Les isotopes émetteurs de positrons sont instables en raison d'un excès d'un proton dans leur noyau. Pour passer à l'état stable, ils se désintègrent par la transformation d'un proton en un neutron accompagnée de l'émission d'un neutrino ν et d'un positron e+. Le positron émis a une énergie cinétique initiale qu'il perd continuellement par des multiples interactions coulombiennes avec les atomes du milieu sur un chemin sinueux. Une fois que le positron atteint l'équilibre thermique du milieu, il interagit avec un électron par une réaction d'annihilation au cours de laquelle la masse de ces deux particules se transforme en deux photons de 511 keV émis dans

Tableau 2.I – Les isotopes les plus utilisés dans les examens TEP et leurs propriétés physiques [19].

Isotopes	^{11}C	^{13}N	^{15}O	^{18}F
Demi-vie (min)	20.4	10.0	2.1	109.8
Énergie cinétique moyenne des e+ (MeV)	0.39	0.49	0.37	0.25
Portée moyenne dans l'eau (mm)	1.1	1.5	2.7	0.6

deux directions quasi-opposées (figure 2.2). L'équation suivante résume ces deux processus :

$$\begin{aligned} {}^{A}_{Z}X &\longrightarrow {}^{A}_{Z-1}X + e^{+} + \nu \\ e^{+} + e^{-} &\longrightarrow 2\gamma \end{aligned} \tag{2.1}$$

Par ailleurs, en plus de la portée du positron, un autre facteur physique intrinsèque qui limite la résolution du système TEP est la non-colinéarité des deux photons de 511 keV émis lors de l'annihilation d'un positron et d'un électron qui n'ont pas totalement perdu son énergie cinétique. La distribution de l'angle de déviation est presque gaussienne avec une largeur à mi-hauteur de 0.5 degré dans l'eau.

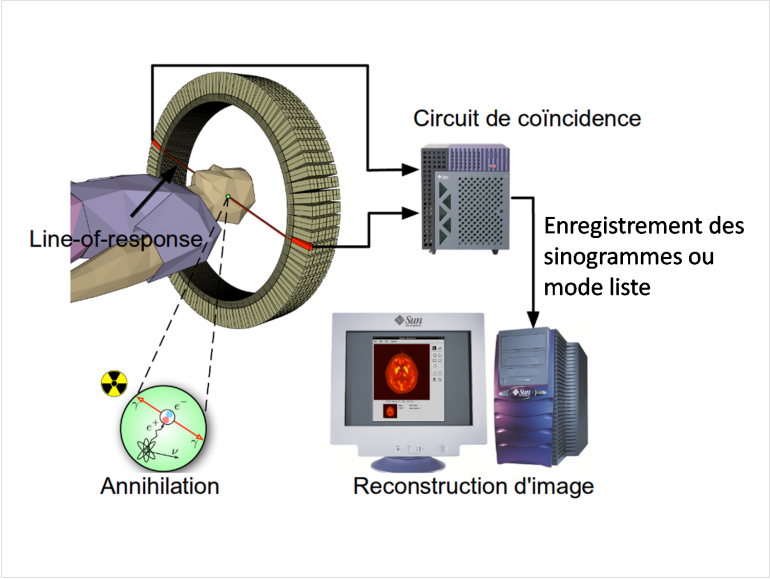


Figure 2.3 – Schéma du principe du système de tomographie d'émission par positrons. Source : Wikipedia, site visité en mai 2015.

2.4 Principe de détection

Le système TEP est constitué d'un anneau, ou de plusieurs anneaux, de détecteurs autour de l'objet. Les détecteurs interceptent les photons de 511 keV provenant de l'annihilation du positron avec l'électron. Les paramètres typiques d'un système de la TEP sont 30 anneaux avec 600 détecteurs chacun. Pour réaliser des projections, chaque paire de détecteurs ne doit considérer et ne comptabiliser que les annihilations qui ont eu lieu le long du tube rectangulaire joignant les deux détecteurs et que l'on appelle LOR. Ceci est réalisé par un circuit de collimation électronique, appelé aussi circuit de coïncidence, qui ne valide que les événements qui correspondent à la détection des deux photons simultanément et dont l'énergie déposée dans chaque détecteur est de 511 keV. En raison de l'imperfection du système de détection, deux critères sont vérifiés en pratique :

1. Les deux photons ont été détectés à l'intérieur d'une fenêtre temporelle de 6 à 15 ns sélectionnée par l'utilisateur. Le choix de la largeur de cette fenêtre est dicté surtout par les scintillateurs inorganiques utilisés pour la détection. En effet, il peut y avoir une différence de temps d'arrivée des deux photons de l'annihilation au circuit de coïncidence due au temps de décroissance du scintillateur et au décalage créé par les circuits électroniques. Plus le temps de décroissance des cristaux est rapide plus les fenêtres temporelles plus courtes sont utlisées.
2. L'énergie déposée par chaque photon dans le détecteur est à l'intérieur d'une fenêtre d'énergie définit autour du 511 keV. La largeur de cette fenêtre dépend de la résolution en énergie des détecteurs et elle est ajustée pour accepter le maximum d'événements sans trop contaminer les données avec du rayonnement diffusé.

À cause de ces deux fenêtres imposées par l'imperfection des systèmes de détection, le circuit de coïncidence laisse passer des faux événements qui sont responsables du bruit dans les mesures (figure 2.4). Ces fausses coïncidences sont notamment :

– **Les coïncidences diffusées**, qui sont les coïncidences détectées lorsque la direction de l'un ou des deux photons d'annihilation a été modifiée par une interaction Compton dans le milieu ou dans les détecteurs. Par conséquent, la ligne de réponse détectée n'est plus reliée à l'emplacement de l'émission. Il en résulte une détérioration du contraste et de la quantification. Une diminution de la largeur de la fenêtre d'énergie permet de diminuer le taux du diffusé, mais entraîne aussi une diminution de la sensibilité du système en raison

de la mauvaise résolution en énergie des détecteurs.

– **Les coïncidences fortuites (aléatoires)**, qui correspondent à la mesure à l'intérieur de la fenêtre temporelle de deux photons issus de deux annihilations différentes. Ces deux photons seraient considérés par le circuit de coïncidences comme provenant d'une annihilation et donc l'événement serait enregistré comme un vrai événement de coïncidence (figure 2.4 d). Ces faux événements qui ne sont pas corrélés à une information spatiale engendrent un bruit statistique de faible fréquence dans l'image reconstruite, et, par conséquent, détériore le contraste au niveau de cette dernière. Le taux de coïncidences aléatoires est une fonction linéaire de la fenêtre temporelle d'acquisition et croît selon le carré de l'activité présente dans le champ de vue. Diminuer la fenêtre temporelle d'acquisition permet de diminuer le taux coïncidences fortuites, mais fait détériorer aussi la sensibilité du tomographe et diminuer donc le RSB pour la même activité.

Un autre facteur physique important qui fausse la quantification est l'atténuation des photons 511 keV dans les tissus du patient par interaction photoélectrique, diffusion Compton et Rayleigh. L'atténuation est un phénomène non isotrope dans l'organisme et elle est fonction de la densité et de l'épaisseur des tissus traversés. La fraction Γ des photons absorbés par les tissus traversés est donnée par

$$\Gamma = 1 - e^{-\sum_i \mu_i x_i} \tag{2.2}$$

où x_i sont les épaisseurs des tissus traversés par les photons et μ_i sont leurs coefficients d'atténuation. Pour un faisceau de 511 keV, ces coefficients d'atténuation sont de l'ordre 0.151 cm^{-1} dans l'os, 0.095 cm^{-1} dans les tissus mous et 0.031 cm^{-1} dans les poumons. Les phénomènes physiques cités ci-dessus tels que les coïncidences aléatoires, les coïncidences diffusées et l'atténuation des photons sont estimés pour chaque acquisition. Des corrections sont effectuées sur les données d'acquisition ou sont intégrées au niveau de l'algorithme de reconstruction. Les livres de Bailey et al. [19] et de Phelps [166] détaillent la correction des différents phénomènes physiques propres au TEP.

2.5 Détecteurs utilisés en TEP

Les détecteurs ont pour rôle détecter les photons provenant de l'annihilation, puis de déterminer l'énergie du photon détecté, l'instant de détection et le lieu de l'interaction. Les performances des systèmes TEP sont très dépendantes des caractéristiques des détecteurs utilisés, telles

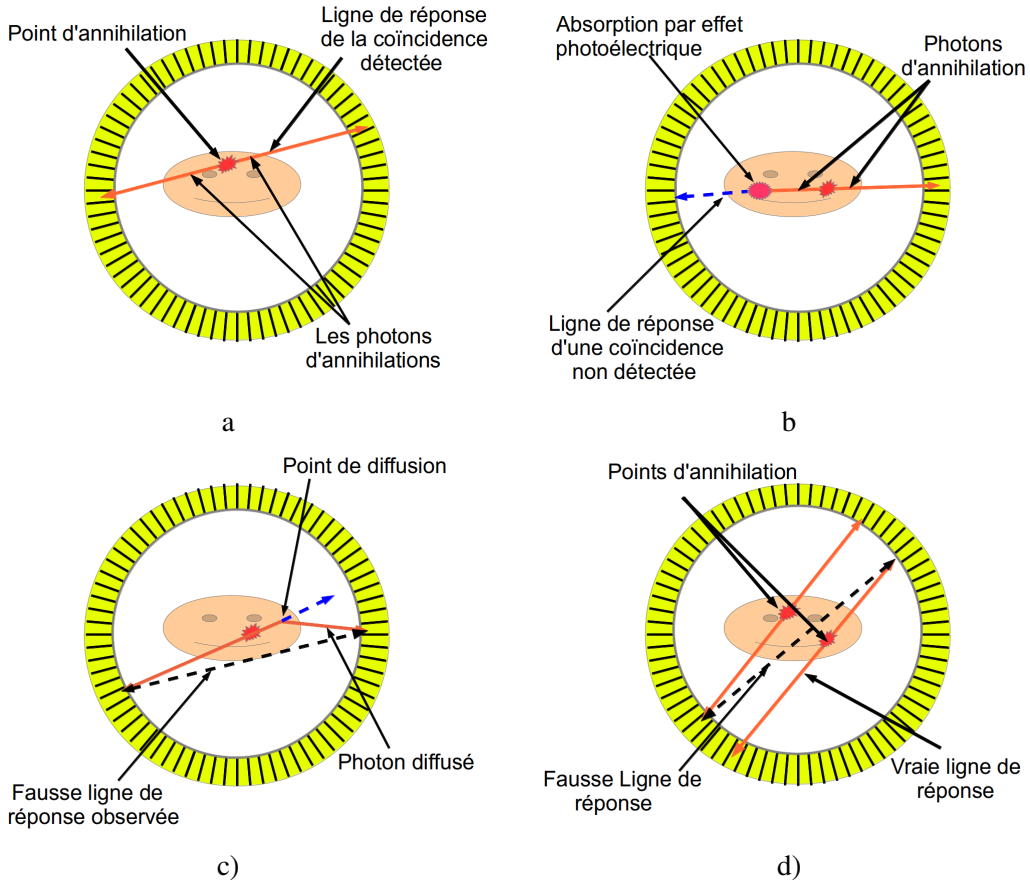


Figure 2.4 – Les coïncidences enregistrées par le système TEP : a) les vraies coïncidences , b) les coïncidences perdues par effet photoélectrique, c) les coïncidences diffusées et d) les coïncidences fortuites.

que l'efficacité de détection des rayons 511 keV, la résolution en énergie et la résolution temporelle. Les détecteurs utilisés en TEP cliniques sont généralement des cristaux inorganiques à scintillation et sont organisés en modules découpés (figure 2.5 b). Le nombre de cristaux par module, ainsi que les dimensions de chaque cristal élémentaire dépendent du manufacturier et du type de système TEP (tableau 2.II). Chaque module est couplé à un nombre limité (ordre de 4) de photomultiplicateurs (PMT). La fonction de chaque cristal est de détecter les photons gamma et de les transformer en un signal lumineux. Ces photons lumineux sont dirigés vers les PMTs qui vont les transformer en un signal électrique (figure 2.5 a). Pour chaque bloc, la détermination du cristal élémentaire où l'interaction a eu lieu s'effectue à partir des signaux fournis par PMT en utilisant le principe de barycentre utilisé dans les caméras à scintillation de type Anger. Le choix

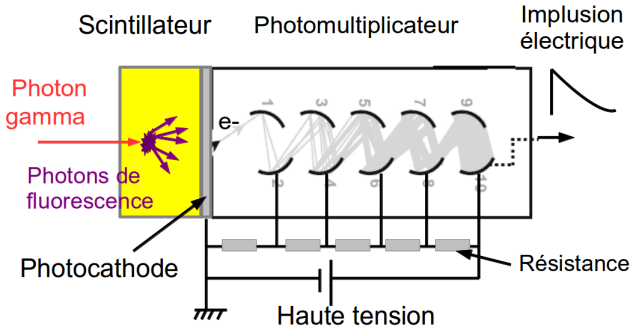
du scintillateur est imposé par plusieurs paramètres physiques, notamment :

- **L'efficacité de détection**. Ce facteur est conditionné par la densité et le numéro atomique effectif Z_{eff} du cristal et il est parmi les paramètres les plus importants dans le choix du détecteur. En effet, il influence directement la sensibilité du système et la résolution spatiale. Un détecteur qui présente un bon pouvoir d'atténuation pour les photons de 511 keV permet de diminuer l'épaisseur de détecteur et donc d'améliorer la résolution spatiale du système.
- **La photofraction**, qui est le pourcentage de photons qui interagissent par effet photoélectrique dans le cristal. Ce facteur est aussi imposé par le Z_{eff} du cristal, car la probabilité de l'interaction photoélectrique est proportionnelle à Z^3_{eff}. Un scintillateur qui présente un bon facteur de photofraction comme le germanate de bismuth (BGO) (tableau 2.III) permet d'améliorer la résolution spatiale en diminuant la diffusion inter-cristaux.
- **Le rendement lumineux**, qui est défini par le nombre de photons lumineux obtenu par absorption d'un rayon gamma. Ce facteur conditionne la résolution en énergie du système et, par conséquent, impose le choix de la largeur de la fenêtre d'énergie d'acquisition, qui elle influence le taux de coïncidences diffusées. Le rendement lumineux est exprimé de manière relative par rapport au rendement de l'iodure de sodium dopé au thallium (NaI (Tl)), car ce dernier possède le meilleur rendement lumineux parmi les types des cristaux à scintillation utilisés en TEP.
- **La constante de décroissance**, qui est un paramètre qui influence la résolution temporelle du détecteur définie par le temps minimal séparant la détection de deux photons que le scintillateur peut différencier. Ce dernier facteur détermine la largeur de fenêtre temporelle d'acquisition à utiliser (2 à 3 fois la résolution temporelle), et par conséquent, elle a un impact direct sur le taux des coïncidences aléatoires. La résolution temporelle typique est 5 à 6 ns pour le BGO et 2 à 3 ns pour le LSO (l'orthosilicate de lutétium). La constante de décroissance a aussi un impact majeur sur le temps mort du système d'acquisition, qui est le temps durant lequel il est impossible de détecter un autre événement que celui qui est en processus de détection.
- **La longueur d'ondes** des photons de scintillation. Il faut que les bandes d'émission et d'absorption du cristal ne se chevauchent pas trop pour éviter que les photons de scintillation ne soient absorbés par le cristal. Par ailleurs, le spectre d'émission doit être compatible avec la réponse spectrale de la photocathode du PMT.

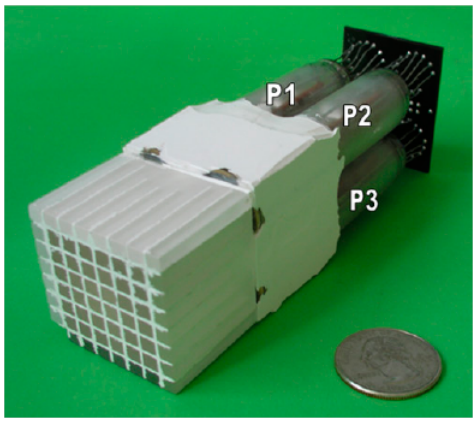
– **L'indice de réfraction**. Il faut que cet indice du cristal et celui de la photocathode du PMT soient compatibles pour permettre une bonne transmission des photons vers la photocathode.

Tableau 2.II – Caractéristiques de quelques système TEP ([15, 29, 106, 213]).

Système TEP	Biograph HIREZ	Discovery RX	Gemini GXL	Gemini TF
Type du cristal	LSO	LYSO	GSO	LYSO
Dimensions du cristal (mm^3)	4x4x20	4.2x6.3x30	4x6x20	4x4x22
Nombre de Cristaux par bloc	13x13	8x6	22x29	23x44
Nombre de cristaux par anneau	624	630	616	644
Nombre d'anneau	39	24	29	44
Champ de vue longitudinale (cm)	58.5	70	56	57.6
Champ de vue axial (cm)	16.2	15.7	18	18



a



b)

Figure 2.5 – a) Schéma de principe de fonctionnement d'un scintillateur couplé un photomultiplicateur, et b) module de détecteurs constitué de 8 x 8 cristaux BGO élémentaires de dimensions 4.0 x 4.1 x 30 mm^3 couplés à 8 photomultiplicateur. Tiré de Pichler et al. [169].

Par ailleurs, d'autres paramètres sont pris en considération dans le choix du cristal, comme l'hygroscopie, l'insensibilité à la température et le coût. Le BGO est le cristal qui était le plus utilisé dans les appareils TEP jusqu'à récemment. Il possède le plus grand coefficient d'atténuation ($\mu = 0.95\ cm^{-1}$) et la plus grande photofraction (0.44). Cependant, il souffre d'une constante de temps relativement longue (300 ns) qui impose l'utilisation d'une fenêtre temporelle de l'ordre de 12 ns et il présente un faible rendement lumineux. Ce matériau a été abandonné pour être

remplacé par des cristaux LSO, GSO (orthosilicate de gadolinium) et par LYSO (orthosilicate de lutécium et d'yttrium dopé au cérium) qui sont légèrement moins denses que le BGO, mais ont une constante de décroissance plus rapide et un bon rendement lumineux. Ces scintillateurs ont permis de diminuer la fenêtre temporelle d'acquisition jusqu'à l'ordre de 600 ps. Ils ont donc permis aux nouveaux systèmes TEP d'intégrer le paramètre du temps de vol dans la reconstruction pour améliorer la quantification. Le principe de reconstruction avec le temps de vol consiste à utiliser la différence de temps d'arrivée des deux photons d'annihilation sur les deux détecteurs pour estimer l'origine de l'annihilation sur la LOR. Le tableau 2.III présente les différents types de cristaux utilisés dans le TEP ainsi que leurs caractéristiques.

Tableau 2.III – Les caractéristiques des cristaux les plus utilisés en TEP. Données tirées de Lewellen [113] et de Phelps [166].

Scintillateurs	**NaI(Tl)**	**BGO**	**LSO**	**GSO**	**LYSO**	**BaF2**
Densité (g/cm^3)	3.67	7.13	7.4	6.7	7.1	4.89
Z_{eff}	51	74	66	59	60	54
Coefficient d'attenuation (cm^{-1})	0.34	0.92	0.87	0.62	0.86	0.44
Indice de réfraction	1.85	2.15	1.82	1.85	1.81	1.56
Rendement lumineux relatif (% NaI(Tl))	100	15	75	30	80	5
Décroissance lumineuse (ns)	230	300	40	65	41	0.8
Résolution en énergie (%)	6.6	10.2	10	8.5	-	11.4
Rapport effets photoélectrique et Compton	0.22	0.78	0.52	0.35	-	0.24

2.6 Acquisition et enregistrement des données

Il existe deux modes d'acquisitions en TEP : le mode bidimensionnel (2D) qui est le mode utilisé dans les premières machines commercialisées et le mode tridimensionnel (3D) qui est le plus utilisé actuellement.

2.6.1 Mode d'acquisition bidimensionnel (2D)

En mode 2D, l'acquisition s'effectue en positionnant des anneaux de plomb ou de tungstène, appelés septas, entre les anneaux de détection. La forme et les dimensions des septas ont été déterminées afin que chaque détecteur ne détecte que les photons ayant une direction presque parallèle au plan transversal du tomographe, c'est dire dans la direction $\theta \simeq 0$; θ étant l'angle azimutal formé par l'incidence des LORs sur le plan transversal droit du tomographe (figure 2.6 a). Les coïncidences ne sont mesurées que par les paires constituées des détecteurs du même anneau afin

de ne réaliser que des coupes selon les plans directs $P_{z,\theta=0}$, où z est le point d'intersection de ce plan avec l'axe Z du tomographe (figure 2.6 a). Par ailleurs, pour augmenter l'échantillonnage axial et la sensibilité du TEP, ce mode permet aussi l'enregistrement des événements simultanés détectés par des paires appartenant à des anneaux adjacents $\theta \simeq 1^\circ$ pour former des plans inclinés. Les plans inclinés voisins seront ensuite combinés pour former des plans droits croisés situés entre les plans directs (figure 2.6 a). Un appareil TEP qui a N anneaux permet un échantillonnage de $2N-1$ coupes transversales : N plans directs et $N-1$ plans croisés. Les algorithmes de reconstruction dans ce mode estiment la distribution du radiotraceur coupe par coupe.

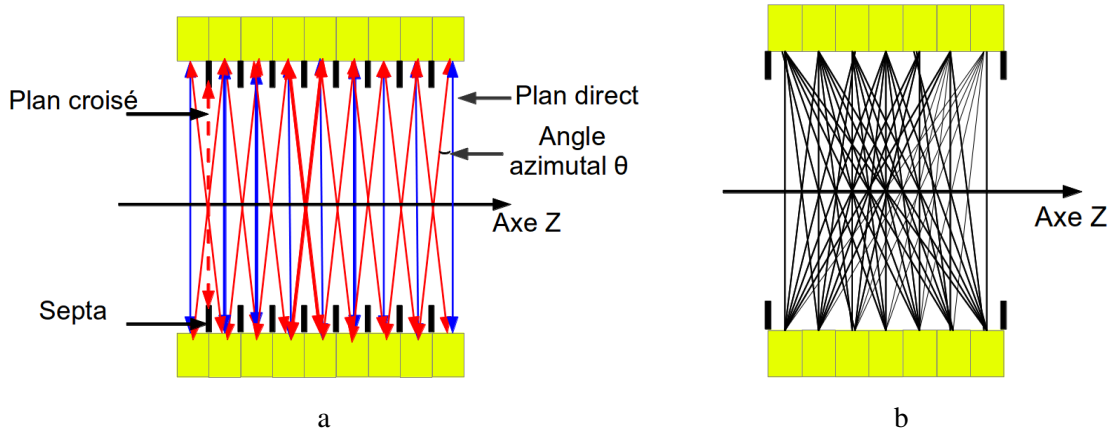


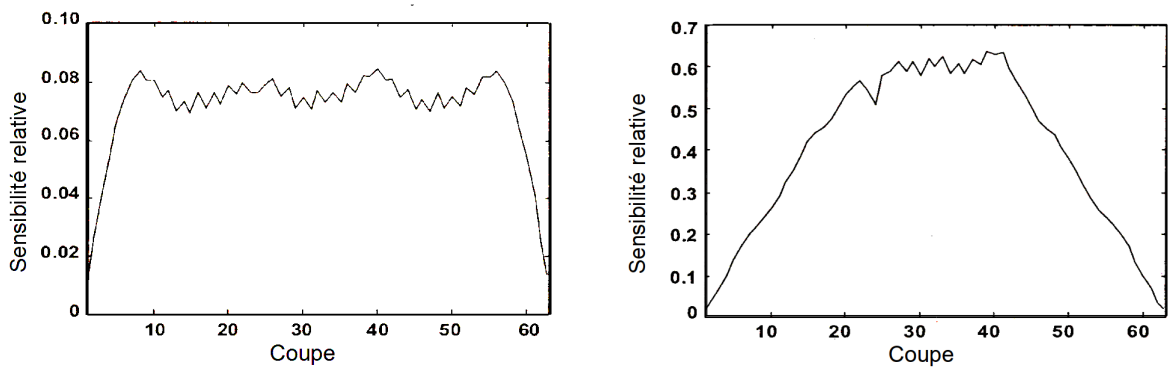
Figure 2.6 – Principes d'acquisition de données par un système TEP de 7 couronnes : a) mode d'acquisition 2D et b) mode d'acquisition 3D.

2.6.2 Mode d'acquisition tridimensionnel (3D)

Le mode d'acquisition 3D est réalisé en rétractant les septas afin de permettre la mesure des événements de coïncidence par l'ensemble des combinaisons de paires de détecteurs (figure 2.6 b). Par rapport au mode 2D, la détection de la coïncidence se fait aussi le long des LORs obliques et permet de réaliser des plans de coupes obliques. Le nombre total de coupes en 3D est donc N^2 mais en pratique la plupart des systèmes utilisent un angle θ maximal pour limiter la quantité des données brutes. Ce mode d'acquisition augmente la sensibilité du TEP par au moins un facteur 5 en comparaison avec le mode 2D [41, 223], ce qui améliore le RSB tout en permettant une diminution du temps d'acquisition et une réduction de la quantité du radiotraceur injecté. Cependant, l'absence des septas entraîne une augmentation importante du taux des événements diffusés

et fortuits. Les coïncidences diffusées peuvent constituer jusqu'à 40% des événements détectés alors que cette proportion est inférieure à 15% en 2D [208]. Cette augmentation des événements diffusés pourrait impliquer une perte significative du contraste par rapport au mode 2D si des méthodes puissantes de correction de ces phénomènes n'étaient pas utilisées [60, 252]. Un autre inconvénient du mode 3D est le nombre de données brutes qui augmente considérablement ; il est de l'ordre de 400 mégaoctets/trame [241]. La gestion et le stockage de ces données, en plus du temps de reconstruction des images, constituent encore des défis dans l'utilisation du mode 3D en milieu clinique.

Le mode 3D a aussi l'inconvénient de produire une variation de la sensibilité le long de l'axe Z (figure 2.7). Le profil de la sensibilité est triangulaire. En effet, le nombre de lignes de réponse qui traverse le patient diminue au fur et à mesure qu'on s'éloigne du centre du tomographe vers les anneaux extérieurs (figure 2.6 b). Pour réduire cette différence de sensibilité et avoir un plateau sur le profil de la sensibilité, des systèmes TEP permettent de sélectionner des angles θ petits lors de l'acquisition. Le choix du θ se fait en spécifiant le paramètre *mrd* (*maximum ring difference*) qui définit la différence maximale entre les couronnes pouvant former des LORs acceptables. Pour uniformiser la sensibilité sur une grande partie de champs, on effectue aussi un recouvrement du champ de vue, typiquement de 25 ou 50%, lorsqu'on déplace la table pour l'exploration des champs étendus.



a b

Figure 2.7 – Variation de la sensibilité dans la direction axiale Z en : a) mode d'acquisition 2D et b) mode d'acquisition 3D. Courbes tirées de Phelps [166].

2.6.3 Enregistrement des données

Les systèmes TEP utilisent deux modes de stockage en mémoire des données d'acquisition. Le premier est le mode histogramme qui est le mode le plus utilisé et qui enregistre les données dans des matrices sinogrammes. Et le deuxième est le mode liste qui stocke les événements par ordre chronologique dans un long fichier binaire (ou texte).

2.6.3.1 Sinogrammes

Le sinogramme $M_{(z,\theta)}(r,\phi)$ est la matrice qui contient l'ensemble des événements de coïncidence qui ont eu lieu dans la coupe $P_{(z,\theta)}$ lors d'une acquisition, où :

- Les indices z et θ sont respectivement la position longitudinale le long de l'axe Z et l'angle azimutal qui forment le plan de coupe (figure 2.9 a)). La positions z est généralement indexée par un entier défini par $z_{i,j} = i + j - 1$ où i et j sont les indices des couronnes qui forment le plan de coupe, et l'angle θ est indexé par $\theta_{i,j} = j - i$. Pour le système TEP de Philips Gemini GXL qui a 29 anneaux et que nous avons utilisé dans ce travail, les valeurs entières qui indexent la position z varient de 0 à 56 et celles qui indexent l'angle θ sont de -28 à 28.
- r est la distance radiale du LOR par rapport au centre de la coupe, et ϕ est l'angle polaire qui détermine la direction de la projection le long du LOR dans le plan de la coupe, appelé aussi vue. En effet, chaque LOR est définie par ses coordonnées polaires (r,ϕ) dans le plan de la coupe (figure 2.8). Un sinogramme possède autant de lignes que le tomographe offre d'angles de vue et autant de colonnes que le nombre de LORs pour une direction ϕ donnée. Typiquement, pour un tomographe ayant M détecteurs par couronne, la taille d'un sinogramme est typiquement de l'ordre $M/2$ lignes (vues) x M colonnes (positions radiales)[201]. Pour chaque vue ϕ, le nombre de colonnes est constitué de $M/2$ LORs qui connectent les deux détecteurs qui sont l'un en face de l'autre (ϕ et $\phi + \pi$) et $M/2$ LORs qui lient les détecteurs quasi opposés (ϕ et $\phi(1+2/M)$) (figure 2.9). Cette stratégie permet d'améliorer l'échantillonnage dans la direction r. Généralement, la distance radiale r est indexée par des entiers r_i qui varient de $-M/2$ à $M/2-1$ et l'angle de vue ϕ est indexé par des entiers ϕ_j dont les valeurs sont de 0 à $M/2-1$. r_i et ϕ_j sont les indices qui adressent les colonnes et les lignes de la matrice sinogramme.

L'information associée à un pixel du sinogramme est la somme des émissions des paires de photons suivant la LOR définie par (r,ϕ) dans le plan du coupe $P_(z,\theta)$. Chaque fois que le circuit de coïncidence accepte un événement sur une LOR, la valeur du pixel du sinogramme qui correspond à cette ligne est incrémentée de 1. Les nouveaux systèmes TEP ont plus de 30 couronnes

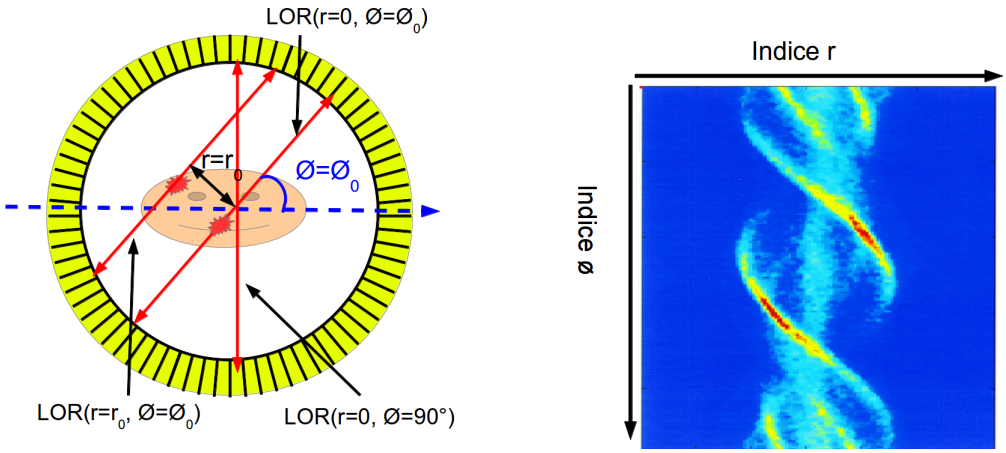
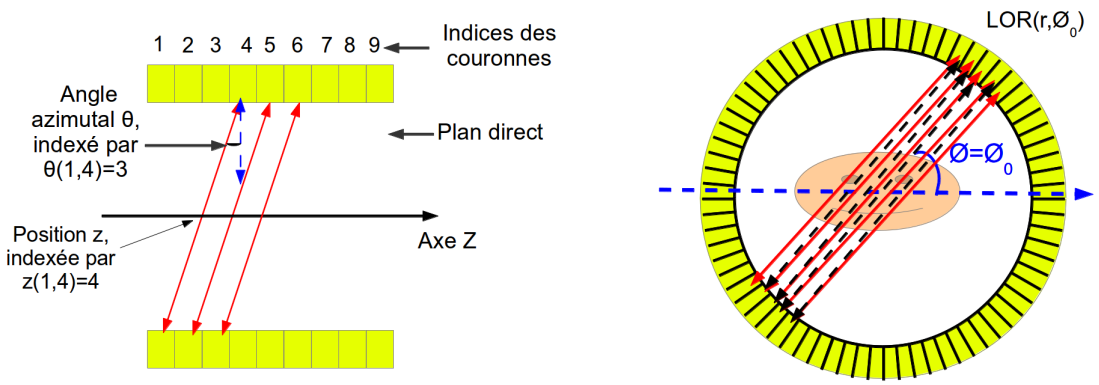


Figure 2.8 – Principe d'enregistrement des données dans un sinogramme.

avec plus de 600 détecteurs élémentaires par couronne. La taille des données d'acquisitions en mode 3D peut atteindre 500 mégaoctets/trame (N^2 sinogrammes x M angles de projection (vues) x $M/2$ LORs par vue x 4 octets : N est le nombre de couronnes et M est le nombre de détecteurs par couronnes). Donc pour réduire cette taille afin d'accélérer la reconstruction, mais sans diminuer la sensibilité en limitant l'angle θ d'acquisition, les manufacturiers utilisent une technique de compression qui consiste à fusionner les sinogrammes voisins. Cette fusion se fait selon un diagramme appelé Michelogramme et qui définit les sinogrammes à fusionner (figure 2.10). Le Michelogramme est paramétré par deux variables qui sont le *mrd* et le *span*. Le *mrd*, paramètre déjà défini, est la différence maximale entre les couronnes, et le *span* est la somme du nombre impair maximal de sinogrammes combinés (pour former un plan direct) et du nombre pair maximal de sinogrammes combinés (pour former des plans croisés). Comme le montre la figure 2.10, le Michelogramme est divisé en un nombre de segments dont chacun est caractérisé par la différence moyenne entre les couronnes et dont le nombre est donné par

$$\text{Nombre de segments} = \frac{mrd+1}{span} \tag{2.3}$$



a) b)

Figure 2.9 – Les coordonnées définissant un sinogramme : a) indices z et θ qui déterminent le plan de coupe et b) indices r et ϕ des lignes de réponses dans un sinogramme : lignes continues lient les détecteurs opposés dans la direction ϕ_0 et les lignes pointillées connectent les détecteurs quasi opposés pour augmenter l'échantillonnage.

2.6.3.2 Mode liste

En mode liste, les événements de coïncidence sont enregistrés un par un en ordre chronologique dans un fichier, généralement, binaire. Les informations enregistrées par événement sont au minimum les suivantes : les indices des deux détecteurs impliqués dans la détection, le temps exact de détection, l'énergie déposée dans le cristal, et le décalage temporel de détection des deux photons. Pour le moment, le mode liste est peu utilisé à cause de la taille des fichiers qui est gigantesque, et du temps de reconstruction qui est très long. Le nombre d'événements d'une acquisition est de l'ordre de 10^8 à 10^{10} pour les tomographes modernes alors que chaque événement est typiquement codé sur 64 bits [241]. De plus, la taille des fichiers et le temps de reconstruction ne sont pas prédéfinis comme dans le cas des sinogrammes, ils croissent avec le nombre d'événements de l'acquisition.

Cependant, pour les acquisitions rapides (dynamiques) et pour les tomographes haute résolution comme ECAT HRRT de Siemens qui a 119 808 détecteurs offrants 4.486 10^9 LORs [49], plusieurs voxels des sinogrammes vont contenir la valeur 0 événement et, par conséquent, la taille du fichier mode liste sera plus petite que la taille du fichier contenant les sinogrammes. Dans ce cas le mode liste serait avantageux en ce qui a trait à la taille de stockage des données et du temps

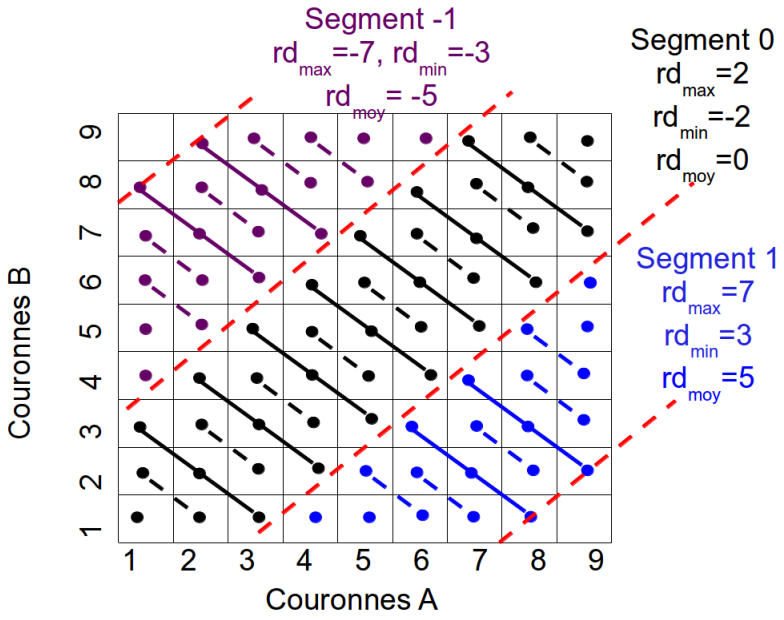


Figure 2.10 – Michelogramme d'un système TEP constitué de 9 anneaux pour un *mrd*=7 et *span*=5. Les lignes continues indiquent les coupes directes et les lignes pointillées indiquent les coupes croisées. $mrd_{\max}$, $mrd_{\min}$ et mrd_{moy} sont respectivement la différence maximale, la différence minimale et la différence moyenne entre les couronnes.

de reconstruction des images.

2.7 Les applications de la TEP

La majorité des systèmes TEP commercialisés actuellement sont des systèmes TEP-TDM qui combinent les deux modalités TEP et TDM. Lors de la réalisation d'un examen TEP, le TDM est utilisé pour obtenir une acquisition faible dose afin de construire une image TDM 3D relativement bruitée en comparaison aux images diagnostiques. Cette image : 1) permet d'estimer la distribution volumique des coefficients d'atténuation qui seront utilisés pour corriger l'effet d'atténuation au niveau des images TEP, et 2) sera fusionnée avec l'image TEP pour permettre une localisation anatomique de la distribution 3D du traceur radioactif fournie par l'image TEP (figure 2.1).

Durant les deux dernières décennies, la modalité TEP-TDM s'est imposée comme référence en oncologie pour le diagnostic, le bilan préthérapeutique, le contrôle de traitement, la différenciation entre tumeurs malignes et tumeurs bénignes, la mise en évidence des tumeurs récidives

et la détection des métastases. Cette réussite est due aussi à la multiplication des centres de cyclotrons pour produire localement le FDG marqué au ^{18}F qui est un isotope émetteur de positrons ayant une demi-vie de 110 minutes. Ce traceur est une molécule analogue du glucose que les cellules cancéreuses, qui sont caractérisées par un métabolisme élevé, vont accumuler beaucoup plus que les cellules saines, permettant ainsi la visualisation du tissu tumoral par le TEP.

La caractérisation des cellules cancéreuses par d'autres radiotraceurs est un domaine de recherche très actif qui va encore asseoir la notoriété des examens TEP en cancérologie. Ainsi, plusieurs nouveaux radiotraceurs commencent à être utilisés et évalués cliniquement, notamment les molécules : ^{11}C-choline utilisée pour quantifier le métabolisme lipidique des tumeurs [165], ^{18}F-FDOPA une molécule analogue au FDOPA et qui permet d'évaluer le transport des acides aminés, en particulier dans le striatum du cerveau humain [23], et ^{18}F-FLT pour mesurer la prolifération cellulaire des tumeurs [72]. Concernant l'utilisation des images TEP-TDM pour délimiter les tumeurs lors de la planification du traitement en radiothérapie, il n'y a pas encore un consensus clinique sur ce sujet [123]. En effet, la quantification des images en TEP dépend énormément du processus d'acquisition, de la calibration et des caractéristiques des systèmes TEP. Cependant, il a été démontré que les images TEP-TDM peuvent améliorer le plan du traitement établi à partir des images TDM dans certains cancers, notamment pour les cancers du poumon non à petites cellules [50] et ceux de la tête et du cou [164]. L'utilisation de la TEP-TDM pour l'estimation des volumes cibles en radiothérapie serait de plus en plus une avenue intéressante avec l'amélioration de la quantification et la standardisation du processus de la réalisation des examens TEP.

La modalité TEP est utilisée aussi en cardiologie et en neurologie. En cardiologie, elle permet surtout d'évaluer la viabilité et de mesurer quantitativement le flux d'irrigation du myocarde. En neurologie, elle permet de diagnostiquer les gliomes, de localiser les foyers épileptogènes et de caractériser des cas de démence comme la maladie d'Alzheimer. Par ailleurs, la mise au point ces dernières années de la modalité qui combine la TEP et l'imagerie par résonance magnétique (IRM) (TEP-IRM) va permettre d'améliorer énormément le diagnostic précoce de plusieurs maladies neurologiques [168]. La TEP est aussi une modalité qui est de plus en plus utilisée dans l'industrie pharmacologique pour évaluer la biodistribution des nouveaux médicaments.

2.8 Conclusion

La TEP est une modalité d'imagerie fonctionnelle capable de quantifier les paramètres biochimiques et physiologiques au niveau cellulaire. Elle permet ainsi de détecter une activité métabolique anormale au niveau moléculaire dans des organes dont la morphologique apparaît normale aux examens effectués avec les autres modalités radiologiques. L'amélioration des performances des systèmes avec le développement des détecteurs plus denses et plus rapides pour augmenter la sensibilité et diminuer le temps d'acquisition, l'introduction des systèmes d'acquisition 3D et la combinaison des systèmes TEP-TDM ont permis d'améliorer l'exactitude de la quantification de l'information fonctionnelle TEP et de combiner l'image anatomique TDM et l'image fonctionnelle pour mieux localiser les tumeurs. La multiplication des centres de cyclotrons pour produire le FDG qui est la molécule la plus utilisée pour quantifier le métabolisme cellulaire a fait ces dernières décennies de la TEP la modalité de référence dans le diagnostic et le suivi des tumeurs.

La quantification en TEP souffre d'imprécision. En effet, on réduit la taille des données par compression, ce qui compromis l'effort qui a été fait au niveau des détecteurs pour réduire leur taille et augmenter leur nombre afin d'améliorer la résolution spatiale et l'exactitude de la qunatification. De plus, le grand potentiel de cette modalité d'offrir des informations cliniques sur le métabolisme, la perfusion, et la prolifération cellulaire, ainsi que sur la densité des récepteurs et expression des gènes afin d'établir des diagnostics précoces et fiables de plusieurs maladies n'est pas encore totalement exploité. Ces limitations sont dues en partie aux approximations faites au niveau des algorithmes de reconstructions pour rendre le temps de reconstruction des images acceptables cliniquement. Par ailleurs, le mode liste n'est pas utilisé en clinique de routine, alors qu'il permet d'utiliser plus efficacement le temps du vol et de corriger l'effet du mouvement afin d'améliorer la résolution spatiale, et qu'il permet d'estimer les paramètres physiologiques avec plus de précision en utilisant le temps de détection des événements. Les modèles mathématiques du système d'acquisition et des données d'acquisition sont aussi simplifiés pour accélérer la reconstruction. Le développement des plates-formes de calcul parallèles moins chères telles que les GPU permettra dans l'avenir d'exploiter tout le potentiel de cette modalité dans le diagnostic des maladies et l'évaluation des réponses à la thérapie.

CHAPITRE 3

MÉTHODES DE RECONSTRUCTION : PROBLÈME DIRECT

La modélisation du système d'acquisition TEP, appelée aussi résolution du problème direct ou projection, consiste à développer un modèle mathématique capable de prédire les mesures du tomographe connaissant la distribution du traceur. La modélisation est une étape essentielle pour la majorité des algorithmes de reconstruction.

L'exactitude des images estimées et le RSB pour ces images dépendent en grande partie de la l'exactitude du modèle. Par ailleurs, un modèle plus exacte est un modèle plus complexe qui se traduit par une augmentation importante du temps de calcul lors de l'exécution de l'algorithme de reconstruction.

3.1 Paramétrisation de l'image objet

La distribution spatio-temporelle du radio-traceur dans de l'objet Ω à explorer peut être définie par une fonction continue $f(r,t)$ représentant la concentration spatiale du traceur en MBq/mL ou mCi/mL en fonction du temps. Le vecteur $r = [x,y,z]$ détermine la position spatiale et le scalaire t est la variable de temps. La reconstruction de la fonction continue $f(r,t)$ à partir des mesures est impossible à cause du nombre limité de ces dernières. De ce fait, la fonction $f(r,t)$ est approximée par une combinaison d'une suite de fonctions $\psi_m(r,t), m = 1...M$:

$$f(r,t) \simeq \sum_{m=1}^{m=M} c_m \psi_m(r,t). \tag{3.1}$$

Le choix de la limite M se fait en fonction des résolutions spatiale et temporelle du système TEP. Si les fonctions $\psi_m(r,t)$ forment une base orthonormée dans l'espace d'Hilbert $4D\ L2(\Omega,T)$ engendré, alors

$$c_m = \int_\Omega \int_T f(r,t)\psi_m(r,t), \tag{3.2}$$

où T est l'intervalle de temps d'acquisition des données. La méthode la plus simple et la plus classique pour définir les fonctions $\psi_m(r,t)$ est de considérer

$$\psi_m(r,t) = \phi_i(r)\varphi_j(t), i = 1...I, j = 1...J \text{ et } IxJ = M. \tag{3.3}$$

L'équation 3.1 s'écrira donc

$$f(r,t) \simeq \sum_{i=1}^{i=I} \sum_{j=1}^{i=J} c_{i,j} \phi_i(r) \varphi_j(t). \tag{3.4}$$

Les suites $\phi_i(r)$ et $\varphi_j(t)$ permettent la paramétrisation de la concentration du traceur respectivement dans l'espace et en fonction du temps. Les fonctions $\phi_i(r)$ les plus utilisées sont les fonctions portes définies en divisant l'espace Ω en plusieurs petits cubes uniformes Ω_i appelés voxels (pixels dans le cas de 2D) puis en prenant

$$\begin{aligned} \phi_i(r) &= 1 \text{ si } r \in \Omega_i, \\ \phi_i(r) &= 0 \text{ ailleurs.} \end{aligned} \tag{3.5}$$

De même les fonctions $\varphi_j(t)$ généralement utilisées sont définies par

$$\begin{aligned} \varphi_j(t) &= 1 \text{ si } t \in [t_{j-1}, t_j[, \\ \varphi_j(t) &= 0 \text{ ailleurs,} \end{aligned} \tag{3.6}$$

où les $[t_{j-1}, t_j[$ sont les intervalles obtenus par discrétisation de l'intervalle total d'acquisition $[0, T]$. L'utilisation de ces fonctions de décomposition permet de réécrire l'équation 3.2 sous la forme

$$c_{i,j} = \int_{[t_{j-1}, t_j[} dt \int_{\Omega} f(r,t) \phi_i(r) dr. \tag{3.7}$$

Chaque coefficient $c_{i,j}$ représente donc la moyenne du radiotraceur sur chaque voxel Ω_i durant chaque intervalle d'acquisition $[t_{j-1}, t_j[$. Ceci facilite l'affichage et l'interprétation des images obtenues après réorganisation de ces coefficients. En effet, les voxels cubiques sont mieux adaptés aux supports d'affichage et les images obtenues forment une suite d'images dynamiques dans le temps $(I_1, I_2, ..., I_M)$ dont chaque image I_j représente la moyenne de la concentration durant l'intervalle $[t_{j-1}, t_j[$ et chaque voxel i d'une image I_j contient la valeur moyenne de la concentration sur le cube élémentaire Ω_i. Par ailleurs, des formes sphériques (blobs) ont été aussi utilisées dans la littérature pour définir les fonctions de base $\phi_i(r)$ numérisant l'image objet dans l'espace [81, 114, 126, 234]. D'autres auteurs ont utilisé ce qu'on appelle les voxels (ou pixels) naturels et qui correspondent à la zone d'intersection entre deux LORs de deux paires de détecteurs [226]. Concernant, les fonctions d'échantillonnage temporel $\varphi_i(t)$, des séries de Fourier et des ondelettes ont été utilisées dans des étapes intermédiaires lors de la reconstruction

des images.

3.2 Matrice système

3.2.1 Définition de la matrice système

Connaissant la distribution du radio-traceur $f(r,t)$ dans l'objet durant une acquisition, le problème direct consiste à modéliser le tomographe pour estimer la moyenne $d_{n,j}$ des coïncidences détectées par la paire n de détecteurs durant le fenêtre d'acquisition temporelle $T_j = [t_{j-1}, t_j]$. Si on suppose que la relation entre $f(r,t)$ et le système de détection est linéaire et que les caractéristiques du tomographe sont invariables dans le temps alors on peut écrire que

$$d_{n,j} = \int_{T_j} dt \int_{\Omega} f(r,t) h_n(r) dr, \tag{3.8}$$

où $h_n(r)$ est la fonction de sensibilité du système (*kernel*) qui représente la réponse de la paire de détecteurs n à une source unitaire située à la position r. $h_n(r)$ est indépendant de la variable temps t car on suppose que les caractéristiques du tomographe sont invariables dans le temps.

Tous les phénomènes physiques qui sont linéaires en fonction de f peuvent théoriquement être modélisés par $h_n(r)$, notamment : la projection géométrique, l'atténuation des photons, la sensibilité des détecteurs, la non-colinéarité des deux photons, la diffusion des photons et leur pénétration dans le cristal, la diffusion inter-cristaux et le diffusé dans le patient. Mais le taux des coïncidences fortuites qui est proportionnel à f^2 [115] ne peut pas être intégré aux coefficients $h_n(r)$, il doit être considéré comme un facteur additif aux mesures. Si on remplace $f(r,t)$ par l'équation de discrétisation 3.4 et puis on considère la fonction $\phi_j(t)$ définie dans l'équation 3.6, l'équation 3.8 du modèle, qui prédit le nombre de coïncidences $d_{n,j}$ détecté par la paire de détecteur n durant l'intervalle d'acquisition T_j, s'écrira

$$d_{n,j} = \sum_{i=1}^{I} c_{i,j} \tau_j \int_{\Omega} \phi_i(r) h_n(r) dr, \tag{3.9}$$

où $\tau_j = t_j - t_{j-1}$ est la durée de la fenêtre d'acquisition T_j. La sous matrice système A_j est défini par

$$\mathbf{A}_j(n,i) = \tau_j \int_{\Omega} \phi_i(r) h_n(r) dr. \tag{3.10}$$

Soient $\mathbf{d}_j = [d_{1,j}, d_{2,j}, ..., d_{N,j}]^T$ le vecteur du nombre des coïncidences détectées par chaque paire

de détecteurs durant l'acquisition j et $\mathbf{c}_j = [c_{1,j}, c_{2,j}, ..., c_{N,j}]^T$ le vecteur des coefficients de décomposition de $f(r,t)$, l'équation 3.9 s'écrira alors :

$$\mathbf{d}_j = \mathbf{A_j}\mathbf{c}_j. \tag{3.11}$$

La matrice $\mathbf{A_j} \in \mathscr{R}^{NxI}$ où N est le nombre de paires de détecteurs (de 10^6 à 10^9) et I est le nombre de voxels (de l'ordre de 10^6). Chaque élément $\mathbf{A}_j(n,i)$ de la matrice $\mathbf{A}_j$ quantifie la contribution du voxel i de l'objet Ω aux coïncidences détectées par la paire n durant l'acquisition j. Pour l'ensemble de fenêtres temporelles d'acquisition, on aura :

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_J \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{A}_J \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_J \end{bmatrix}$$

Si on considère que la durée d'acquisition est la même pour toutes les acquisitions j alors $A_j(n,i) = A = \tau \int_\Omega \phi_i(r) h_n(r) dr$. Par la suite, nous considérerons, par souci de simplification de notation, qu'une seule fenêtre d'acquisition (acquisition statique) ce qui permettra d'écrire l'équation 3.11 sous la forme

$$\mathbf{d} = \mathbf{A}\mathbf{c}, \tag{3.12}$$

où $\mathbf{A}$ est la matrice système (MS) qui modélise le processus d'acquisition. Pour rendre le modèle plus réaliste, on doit tenir compte des phénomènes physiques qui n'ont pas été intégrés dans $\mathbf{A}$, notamment les coïncidences fortuites et le diffusé dans le patient. Le diffusé peut être intégré dans $\mathbf{A}$, mais il est généralement modélisé comme un bruit additionnel aux mesures par souci de ne pas complexifier la matrice $\mathbf{A}$. Le modèle sera donc

$$\mathbf{d} = \mathbf{A}\mathbf{c} + \mathbf{r} + \mathbf{s} + \mathbf{n}, \tag{3.13}$$

où $\mathbf{r}$ est le vecteur qui quantifie les coïncidences fortuites, $\mathbf{s}$ est celui qui quantifie les événements de diffusion et le vecteur $\mathbf{n}$ est le bruit qui prend en compte le caractère stochastique du processus d'acquisition. $\mathbf{n}$ est en généralement considéré comme un bruit blanc de moyenne nulle et dont la densité de probabilité est inconnue. Cependant, de cette modélisation déterministe qui modélise le bruit stochastique indépendamment des mesures, il résulte des algorithmes de re-

construction incapables de contrôler le niveau du bruit lors du processus de reconstruction. Les images obtenues sont donc trop bruitées et nécessitent un post-filtrage du bruit aux dépens de l'exactitude de la solution. Par conséquent, des modèles stochastiques plus réalistes conduisant à des algorithmes statistiques plus exactes ont été utilisés ([205]). Les mesures sont considérées comme une variable aléatoire qui obéit à la statistique de Poisson ou de Gauss pour des acquisitions effectuées avec un grand nombre d'événements. L'équation 3.13 du modèle s'écrira donc

$$\mathbf{d} = E(\mathbf{D}) = \mathbf{A}\mathbf{f} + \mathbf{r} + \mathbf{s}, \tag{3.14}$$

où le vecteur de mesures calculées $\mathbf{d}$ est la valeur moyenne de la variable aléatoire $\mathbf{D}$. Les coefficients $\mathbf{A}_{i,j}$ de la matrice système $\mathbf{A}$ désignent les probabilités de détection par la paire de détecteurs i d'une désintégration survenue au voxel j et le vecteur $\mathbf{f}$ est la valeur moyenne de la variable aléatoire $\mathbf{F}$ représentant la densité des désintégrations dans l'objet.

3.2.2 Détermination de la matrice système

La MS est gigantesque pour les tomographes 3D (ordre des To), et elle est creuse et présente plusieurs symétries. Ces propriétés sont généralement exploitées pour réduire énormément sa taille et accélérer sa création [77, 195, 253]. La MS devient dense et le nombre des symétries diminue au fur et à mesure qu'elle tient compte de différents phénomènes physiques pour augmenter la précision du modèle dans le but d'améliorer la qualité des images reconstruites.

La MS est généralement calculée en temps réel (*on-the-fly*) durant la reconstruction par des méthodes analytiques simples. Parmi ces méthodes, on cite l'algorithme de Siddon (*Siddon ray tracing*) ([77, 209, 254]) qui définit le coefficient $A_{i,j}$ comme la longueur de l'intersection (LOI : *length-of-intersection*) de la ligne joignant les deux centres de la paire de détecteurs i avec le voxel j. Pour améliorer l'échantillonnage du volume, surtout lorsque la taille des voxels est inférieure à la largeur des détecteurs, certains auteurs [139] ont utilisé l'algorithme multi-trajectoires (*multi-tracing*) qui considère plusieurs lignes joignant les paires de détection. D'autres auteurs ont considéré le volume d'intersection (VOI : *volume-of-intersection*) entre le voxel j et le tube qui lie les deux détecteurs i [97, 156, 170, 196].

Des méthodes de calcul analytiques complexes pour améliorer la précision de la modélisation ont été développées [185]. Ces méthodes tiennent compte surtout des facteurs physiques qui introduisent un flou dans l'image et détériorent la résolution tels que la portée du positron, la

non-colinéarité des deux photons, la diffusion des photons et leur pénétration dans les cristaux et la diffusion inter-cristaux. Cependant, à cause de l'intensité de calcul lors de la reconstruction, la MS est généralement calculée en temps réel durant la reconstruction en utilisant des méthodes analytiques très simples et rapides. Les autres effets physiques sont précorrigés sur les données d'acquisition au lieu d'intégrer ces corrections dans les algorithmes de reconstruction, ce qui détériore l'exactitude de la quantification.

Pour améliorer l'exactitude de la MS, des auteurs ont essayé aussi de la mesurer avec une source ponctuelle [6, 159, 221] et de l'enregistrer pour être utilisée, après, dans la reconstruction. Cependant, cette approche constitue non seulement un défi pour la mesurer avec précision dans le cas des tomographes 3D, mais à cause de la taille de cette matrice, le temps d'accès aux éléments de la matrice serait très long et, par conséquent, la reconstruction serait aussi très longue par rapport aux contraintes cliniques.

L'autre méthodologie utilisée pour précalculer la MS est la simulation Monte Carlo [4, 111, 181, 207, 253]. Cette approche permet une meilleure modélisation du système d'acquisition en tenant compte des phénomènes physiques linéaires. Le problème est que la taille de cette matrice est très grande pour les TEPs 3D modernes même si elle est compressée en utilisant les symétries et en n'enregistrant que les coefficients non nuls [253]. Par conséquent, le temps de reconstruction des images est relativement long pour permettre une utilisation en clinique de cette approche. Néanmoins, la disponibilité des logiciels Monte Carlo tels que GATE [90] et PET-EGS [37] permettant de modéliser facilement les systèmes TEPs, et celle des supercalculateurs facilitent le calcul de la MS dans un temps raisonnable (moins de 5 jours) [253].

Pour combiner les méthodes de calcul citées ci-dessus afin d'améliorer l'exactitude du modèle, tout en diminuant la taille de stockage de la MS en réduisant le temps de calcul, une approche proposée consiste à décomposer $\mathbf{A}$ en produit d'un ensemble de matrices creuses qui peuvent être compressées facilement ([108, 140, 141]) :

$$\mathbf{A} = \mathbf{A}_{det.sens}\mathbf{A}_{det.blur}\mathbf{A}_{att}\mathbf{A}_{geom}, \tag{3.15}$$

où :

– $\mathbf{A}_{geom}$ est une matrice de taille IxJ avec I le nombre de LORs et J le nombre de voxels, et qui représente la probabilité géométrique pour qu'une paire de photons produite dans le voxel j atteigne la face de la paire de détection i. $\mathbf{A}_{geom}$ est généralement calculée par la

méthode du trajectoire (*ray tracing*) ou du volume d'intersection. Pour des modèles plus précis, $\mathbf{A}_{geom}$ tient compte aussi des angles solides sous lesquels deux détecteurs qui forment la paire sont vus par le voxel, et du parcours moyen des positrons avant l'émission des photons d'annihilation. $\mathbf{A}_{geom}$ est une matrice creuse qui présente en même temps plusieurs symétries. Donc, elle peut être calculée efficacement en temps réel par des méthodes analytiques en exploitant les différentes symétries afin d'accélérer la reconstruction. Elle est, en effet, la matrice la plus sollicitée dans l'exécution des algorithmes de reconstruction : multiplication par $\mathbf{A}_{geom}$ pour calculer des projections et multiplication par $\mathbf{A}_{geom}^{T}$ pour réaliser des rétroprojections. $\mathbf{A}_{geom}$ peut aussi être précalculée avec précision par les méthodes analytiques et stockée avec une grande compression, mais sa taille reste toujours considérable pour les TEPs 3D.

- $\mathbf{A}_{att}$ est une matrice diagonale de taille IxI dont chaque élément $\mathbf{A}_{att}(i,i)$ de diagonale est le coefficient de correction d'atténuation correspondant à la ligne de réponse i. Cette matrice est calculée à partir de l'image des coefficients d'atténuation des photons d'annihilation dans le patient. Pour les systèmes TEP/TDM combinés, cette image est construite à partir d'une acquisition tomodensitométrique réalisée sur le patient.
- $\mathbf{A}_{det.blur}$ est une matrice de taille aussi IxI et qui modélise la perte de résolution causée par la diffusion des photons dans les cristaux et par l'effet de pénétration inter-cristaux. Elle est soit précalculée par Monte Carlo (Qi et al. [180]) en modélisant statistiquement les propriétés des cristaux, soit précalculée analytiquement [185, 212], ou bien elle est prémesurée [5, 159, 188]. Elle est généralement enregistrée sous forme des sinogrammes pour présenter l'effet de pénétration dans les cristaux et inter-sinogrammes pour modéliser l'effet de la diffusion inter-cristaux.
- $\mathbf{A}_{det.sens}$ est une matrice diagonale IxI qui quantifie l'efficacité de détection de chaque paire de détection. Elle est généralement mesurée directement [76] ou par la méthode de décomposition des facteurs qui combine les mesures et les calculs analytiques [16, 18, 153].

3.3 Conclusion et discussion

La résolution du problème direct consiste à calculer la matrice système qui modélise mathématiquement la chaîne d'acquisition, et qui est utilisée dans les algorithmes de reconstruction

des images TEP afin d'effectuer la projection des données d'acquisition dans l'espace image et pour effectuer la rétroprojection des images dans l'espace des données d'acquisition. Dans ce chapitre, nous avons expliqué les différentes méthodes utilisées pour échantillonner les objets dans l'espace image et puis nous avons défini la MS liant les données mesurées et l'objet image. Cette dernière, qui est une matrice très grande pour les systèmes d'acquisition TEP 3D modernes, est aussi creuse et présente plusieurs symétries. Les algorithmes de reconstruction exploitent ces caractéristiques pour diminuer sa taille et accélérer la reconstruction.

En outre, la résolution du problème est encore problématique dans la reconstruction TEP. En effet, l'exactitude de la modélisation est un compromis entre le temps de reconstruction des images et la qualité de ces images ; plus le modèle devient exacte en tenant compte dans la matrice système de tous les phénomènes physiques d'acquisition, plus les images reconstruites présentent un bon RSB et une bonne résolution spatiale, mais plus aussi la MS devient dense et présente moins de symétries. Par conséquent : 1) le calcul des coefficients de cette dernière devient intense pour les algorithmes qui les calculent en temps réel, et 2) la durée d'accès à la mémoire de l'ordinateur pour chercher ces coefficients ralentit énormément la reconstruction pour les algorithmes utilisant une MS précalculée.

Malgré que des méthodes analytiques plus précises aient été développées pour calculer la MS, que la puissance de calcul disponible actuellement permette de la précalculer par Monte Carlo en simulant tous les phénomènes physiques, et qu'elle puisse aussi être mesurée avec exactitude. L'approche la plus utilisée en clinique consiste à réduire la taille de cette matrice par la combinaison des sinogrammes d'acquisition 3D, à la calculer en temps réel en utilisant des méthodes analytiques très simples et rapides, puis à précorriger les autres effets physiques sur les données d'acquisition au lieu d'intégrer ces corrections dans les algorithmes de reconstruction.

Les GPUs seraient, dans le futur proche, la solution la moins onéreuse pour calculer en temps réel la MS par les méthodes analytiques élaborées ou par Monte Carlo [74]. Puisque la mémoire globale de ces cartes est de plus en plus grande (12 Go pour la Tesla K40X) et que ces derniers présentent de plus en plus des mémoires caches permettant d'accélérer l'accès à la mémoire globale, la reconstruction en GPU en utilisant la MS précalculée est aussi une approche à explorer.

CHAPITRE 4

MÉTHODES DE RECONSTRUCTION : PROBLÈME INVERSE

La reconstruction tomographique consiste à estimer, à partir des données mesurées par le système TEP, la distribution volumique de la concentration de radiotraceur injecté au patient. L'estimation se fait sur des coupes axiales volumiques (coupes tomographiques 2D) ou en volume (reconstruction 3D). On reconstruit aussi à partir des acquisitions dynamiques des images 4D représentant la concentration du radiotraceur en fonction du temps. Ces images permettent d'estimer par un post-traitement les paramètres pharmaco-cinétiques fonctionnels des tissus ou des organes ciblés par le radiotraceur. Par ailleurs, des travaux de recherche ont été effectués ces dix dernières années afin d'estimer les paramètres pharmaco-cinétiques directement à partir des données mesurées par le système TEP.

La théorie de la reconstruction des images à partir des projections a été développée par Radon en 1917. Dans son travail, Radon a démontré qu'un objet en 2D (3D) peut être reconstruit exactement à partir de l'ensemble des projections 1D (2D). En imagerie médicale, la reconstruction tomographique, appelée aussi problème inverse, est un problème mal posé, car le nombre de projections est limité (le nombre des LORs est fini en TEP) et les données acquises sont bruitées. La solution n'est pas unique et elle est instable. La paramétrisation du problème direct facilite la recherche d'une solution sans pour autant rendre le problème bien posé.

Différentes méthodes se basant sur des approches différentes ont été utilisées pour la résolution du problème inverse en TEP. Les plus connues sont les méthodes analytiques, les méthodes itératives déterministes et les méthodes itératives stochastiques qui estiment l'image à partir des données d'acquisition stockées sous forme d'histogramme (sinogrammes).

Ce chapitre présente une bibliographie des algorithmes de reconstruction utilisés en TEP et explique d'une manière détaillée leur principe. L'originalité de ce chapitre est que pour les algorithmes itératifs qui sont les techniques de reconstruction les plus utilisées en TEP depuis les années 90, le problème inverse est présenté en séparant explicitement le critère de convergence des techniques numériques utilisées pour minimiser ou maximiser ce critère. Cette approche permet de bien saisir la classification de ces algorithmes en méthodes déterministes et stochastiques.

4.1 Méthodes de reconstruction analytiques

4.1.1 Réarrangement des données (*rebinning*)

Comme il a été expliqué au paragraphe 2.6.2, la reconstruction des images volumiques à partir des données mesurées en mode d'acquisition 3D a l'avantage d'augmenter la sensibilité du tomographe et d'améliorer le RSB. En contrepartie, les algorithmes de reconstruction 3D nécessitaient des temps de calcul très longs par rapport aux exigences cliniques, à cause du nombre très élevé de données brutes : N_z x N_z sinogrammes,où N_z est le nombre d'anneaux de détecteurs.

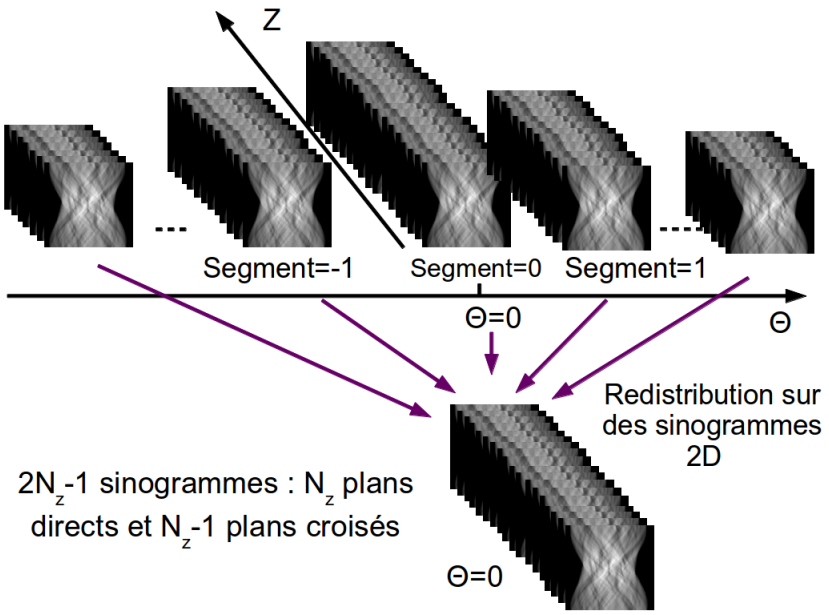


Figure 4.1 – Principe des algorithmes de réarrangement des sinogrammes mesurés en mode 3D pour différents angles θ (segment) en un ensemble de sinogrammes droits $\theta = 0$.

Pour remédier à ce problème, des méthodes de reconstruction 3D approximatives ont été développées. Elles se basent sur une approche qui consiste à redistribuer des données des sinogrammes obliques ($\theta \neq 0$) sur des sinogrammes droits ($\theta = 0$)(figure 4.1) et une reconstruction, par la suite, de chacune des coupes transversales à partir de ces nouveaux sinogrammes droits. La distribution 3D du traceur est obtenue par le regroupement de ces coupes. Cette approche permet de bénéficier des avantages du mode d'acquisition 3D tout en utilisant des algorithmes de reconstruction 2D qui sont plus rapides et mieux maîtrisés.

Ainsi, plusieurs algorithmes ont été développés successivement pour réaliser ce réarrangement des données. Le plus simple est le SSRB (*Single Slice Rebinning*) [47]. Son principe consiste à projeter une LOR oblique dans un plan de coupe droit qui se situe à mi-chemin entre les deux détecteurs correspondants à cette ligne. Cet algorithme simple et rapide entraîne des artéfacts importants dans le cas des acquisitions 3D avec une grande ouverture (grand θ) et pour des sources excentrées. Pour améliorer la précision de l'algorithme SSRB, Lewitt et al. [116] ont proposé la méthode MSRB (*Multi Slice Rebinning*) qui répartit la LOR sur les sinogrammes droits se trouvant entre les deux détecteurs. L'inconvénient de MSRB est son instabilité en présence de bruit. L'algorithme de réarrangement le plus sophistiqué et le plus utilisé est le FORE (*Fourier rebinning*) [51]. À la différence des deux méthodes précédentes, la localisation du sinogramme droit s'effectue dans l'espace fréquentiel. Le principe consiste à calculer la transformée de Fourier de chacun des sinogrammes 2D obliques, et d'appliquer le principe fréquence-distance pour déterminer la position z du plan de coupe droit de projection [38].

L'algorithme FORE combiné avec la méthode de reconstruction itérative OSEM ou à la méthode de rétroprojection filtrée 2D sont les techniques de reconstruction les plus utilisées en TEP. Par ailleurs, les méthodes de reconstruction 3D approximatives qui se basent sur la redistribution des données ont tendance à créer des distorsions spatiales et à amplifier le bruit statistique. Avec le développement des ordinateurs puissants, ces méthodes de reconstruction 3D approximatives ont tendance à être délaissées pour des reconstructions 3D directes.

4.1.2 Méthodes analytiques 2D

Les méthodes analytiques 2D ont été développées vers le début des années 70 pour la reconstruction tomographique en tomodensitométrie. Elles ont été utilisées par la suite, vers la fin des années 70 et le début des années 80, pour la reconstruction des images en tomographie d'émission de monophotonique et en tomographie par émission de positrons à partir des acquisitions 2D ou des acquisitions 3D réarrangées en sinogrammes 2D. Ces méthodes se fondent sur le théorème de la tranche centrale.

4.1.2.1 Théorème de la tranche centrale

En 2D, la transformée de Radon $p(x_r, \phi)$ de la distribution de traceur $f(x,y)$ est la projection de cette distribution le long des lignes de réponse. Pour un angle ϕ donné, $p(x_r, \phi)$ est une droite

donnée par

$$p(x_r,\phi) = \int_{-\infty}^{+\infty} f(x,y)dy_r, \tag{4.1}$$

où (x_r, y_r) sont les coordonnées dans le nouveau repère obtenu par rotation d'un angle ϕ du repère original :

$$\begin{bmatrix} \mathbf{x}_r \\ \mathbf{y}_r \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{4.2}$$

où (x,y) sont les coordonnées dans le repère original. Soient $F(\mu,\nu)$ la transformée de Fourier 2D de $f(x,y)$ et $P_1(\mu_r,\phi)$ la transformée de Fourier 1D sur x_r de $p(x_r,\phi)$:

$$\begin{aligned} F(\mu,\nu) = TF_2 f(x,y) &= \int_{-\infty}^{+\infty} f(x,y)exp(-2i(\pi(x\mu+y\nu))dxdy, \\ P_1(\mu_r,\phi) = TF_2 p(x_r,\phi) &= \int_{-\infty}^{+\infty} p(x_r,\phi)exp(-2i\pi(x\mu_r)dx_r. \end{aligned} \tag{4.3}$$

Le théorème de la tranche centrale stipule que la transformée de Fourier 1D sur x_r de $p(x_r,\phi)$ coïncide avec la droite radiale suivant l'angle ϕ passant par l'origine de la transformée de Fourier 2D de la fonction $f(x,y)$ (figure 4.2), c'est à dire

$$P_1(\mu_r,\phi) = F(\mu,\nu)|_{\mu=\mu_r\cos\phi,\ \nu=\mu_r\sin\phi}, \tag{4.4}$$

où (μ,ν) sont les coordonnées dans le repère fréquentiel original.

4.1.2.2 Rétroprojection simple

L'algorithme analytique de base est la superposition linéaire de rétroprojection (épandage) des valeurs de la projection (figure 4.3). Elle consiste à incrémenter par la valeur de la projection chaque pixel de l'image qui se trouve sur la LOR correspondante à cette projection :

$$f'(x,y) = \int_0^{\pi} p(x_r,\phi)d\phi = \int_0^{\pi} p(x\cos\phi + y\sin\phi,\phi)d\phi. \tag{4.5}$$

Numériquement, cette équation est équivalente à :

$$f'(x,y) = \frac{1}{N}\sum_{n=1}^{n=N} p(x_r,\phi_n) = \frac{1}{N}\sum_{n=1}^{n=N} p(x\cos\phi_n + y\sin\phi_n,\phi_n), \tag{4.6}$$

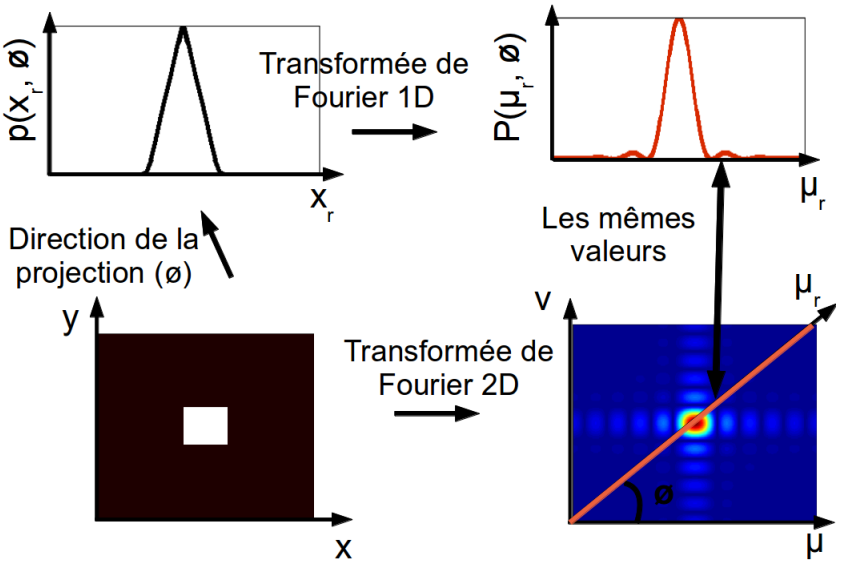


Figure 4.2 – Schéma expliquant le principe du théorème de la tranche centrale.

où N est le nombre d'angles de projection. Au fur et à mesure qu'on rétroprojette les données des projections, les valeurs des zones contenant l'activité augmentent. Cependant, des artéfacts se forment à cause de la persistance des résidus de l'épandage dans toute la coupe, même dans les zones où il n'y a pas d'activité (figure 4.3). Ces artéfacts forment une composante continue qui déforme énormément l'image. En effet, on démontre que :

$$f'(x,y) = f(x,y) \bigotimes \frac{1}{\sqrt{x^2+y^2}}, \tag{4.7}$$

où $\bigotimes$ désigne l'opération du produit de convolution. Cette équation se traduit dans le domaine fréquentiel par

$$F'(\mu,\nu) = F(\mu,\nu)\frac{1}{\sqrt{\mu^2+\nu^2}}. \tag{4.8}$$

La fonction $h(x,y) = 1/\sqrt{x^2+y^2}$, appelée fonction de dispersion ponctuelle, a pour effet de rendre l'image plus floue en amplifiant les faibles fréquences et en atténuant les hautes fréquences.

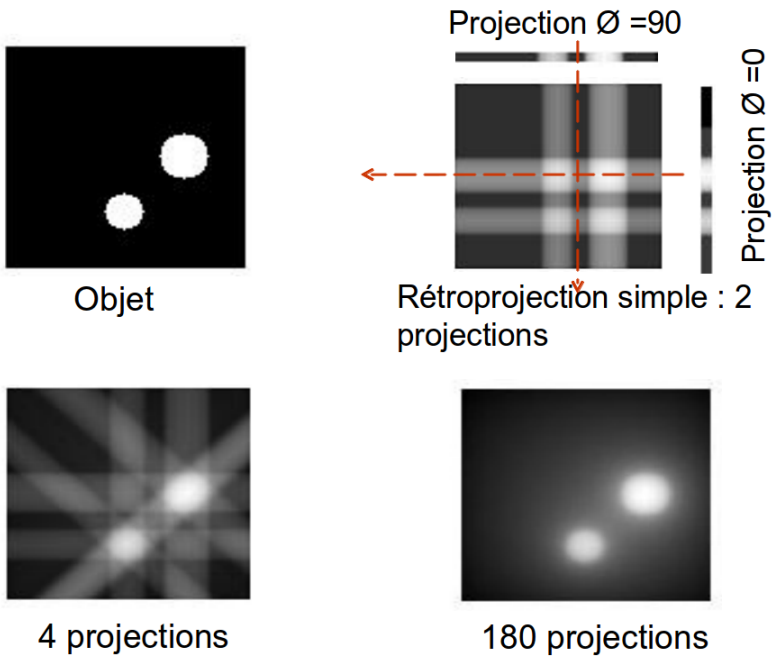


Figure 4.3 – Schéma expliquant le principe de la rétroprojection simple.

4.1.2.3 Reconstruction par méthode directe de Fourier

La reconstruction tomographique par méthode directe de Fourier est une application directe du théorème de la tranche centrale. Elle consiste à :

1. Calculer pour chaque angle ϕ la transformée de Fourier 1D de la projection correspondante : $P_1(\mu_r,\phi) = TF_1 p(x_r,\phi)$.
2. Reporter les échantillons de chaque transformée de Fourier 1D $P_1(u_r,\phi)$ dans le plan spectral cartésien 2D(μ,ν), selon le théorème de la tranche centrale :
   $P_1(\mu_r,\phi) = F(\mu,\nu)|_{\mu=\mu_r\cos\phi,\ \nu=\mu_r\sin\phi}$.
3. Interpoler pour déterminer les composantes spectrales $F(\mu,\nu)$ au centre de chaque pixel du plan spectral cartésien 2D(μ,ν).
4. Calculer la transformée de Fourier inverse 2D de $F(\mu,\nu)$ pour estimer $f(x,y)$. Le problème majeur avec cette méthode est que les transformées de Fourier 1D $P_1(\mu_r,\phi)$ sont échantillonnées d'une manière radiale dans le plan spectral cartésien 2D(μ,ν) (figure 4.4). Il est donc nécessaire d'interpoler pour estimer dans ce plan cartésien la transformée de Fourier

$F(\mu,\nu)$. Par ailleurs, l'image estimée est très sensible aux erreurs de l'interpolation alors qu'une interpolation précise augmente beaucoup le temps d'exécution de l'algorithme.

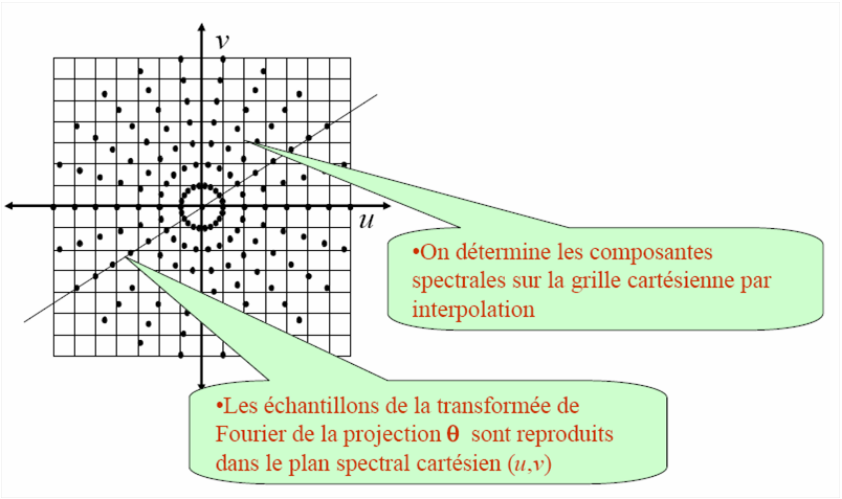


Figure 4.4 – La transformée de Fourier 2D obtenue à partir de la transformé de Fourier 1D des projections. Schéma tiré du cours d'imagerie médicale de M. Bertrand. École polytechnique de Montréal, 2005.

4.1.2.4 Rétroprojection filtrée

La transformée de Fourier inverse nous permet d'écrire que

$$f(x,y) = TF_2^{-1}F(\mu,\nu) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(\mu,\nu)exp(i2\pi(x\mu+y\nu))d\mu d\nu. \tag{4.9}$$

Par un changement de coordonnées cartésiennes en coordonnées en polaires $(\mu,\nu)\rightarrow(\upsilon,\phi)|(0\leq\upsilon\leq+\infty,\ 0\leq\phi\leq 2\pi)$, on pbtient :

$$f(x,y) = TF_2^{-1}F(\mu,\nu) = \int_{0}^{2\pi}\int_{0}^{+\infty} F(\upsilon,\phi)exp(i2\pi\upsilon(x\cos\phi+y\sin\phi))\upsilon d\upsilon d\phi. \tag{4.10}$$

En procédant à un changement des bornes d'intégration, on peut écrire que

$$f(x,y) = \int_{0}^{\pi}\int_{-\infty}^{+\infty} F(\upsilon,\phi)exp(i2\pi\upsilon(x\cos\phi+y\sin\phi))|\upsilon|d\upsilon d\phi. \tag{4.11}$$

Puisque $x_r = x\cos\phi + y\sin\phi$ et que les variables υ et μ_r sont devenues équivalentes après ce dernier changement des bornes d'intégration, alors

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{+\infty} F(\mu_r,\phi) exp(i2\pi\mu_r x_r)|\mu_r| d\mu_r d\phi. \tag{4.12}$$

Or d'après le théorème de la tranche centrale, nous avons : $F(\mu_r,\phi) = P_1(\mu_r,\phi)$, donc

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{+\infty} |\mu_r| P_1(\mu_r,\phi) exp(i2\pi\mu_r x_r) d\mu_r d\phi. \tag{4.13}$$

L'intégrale interne $\hat{p}(x_r,\phi) = \int_{-\infty}^{+\infty} |\mu_r| P_1(\mu_r,\phi) exp(i2\pi\mu_r x_r) d\mu_r$ est la transformée de Fourier inverse de la transformée de Fourier de la projection multipliée par $|\mu_r|$. La quantité résultante $\hat{p}(x_r,\phi)$ est la projection filtrée par le filtre rampe $|\mu_r|$ et la quantité $f(x,y) = \int_0^{\pi} \hat{p}(x_r,\phi) d\phi$ est la rétroprojection de la projection filtrée. De cette dernière équation 4.13 résulte la méthode de reconstruction appelée par la rétroprojection filtrée (FBP : *filtred backprojection*) et dont les étapes d'implantation sont :

1. Calculer pour chaque ϕ la transformée de Fourier 1D de la projection correspondante : $P_1(\mu_r,\phi) = TF_1 p(x,\phi)$.
2. Filtrer dans le domaine fréquentiel chaque projection par le filtre $|\mu_r|$: $\hat{P}(\mu_r,\phi) = |\mu_r| P_1(\mu_r,\phi)$.
3. Calculer la transformée de Fourier inverse de chaque projection filtrée : $\hat{p}(x_r,\phi) = TF^{-1}\hat{P}(\mu_r,\phi)$.
4. Rétroprojetter les projections filtrées : $f^*(x,y) = \sum_{n=1}^{n=N} \hat{p}(x_r,\phi_n) = \sum_{n=1}^{n=N} \hat{p}(x\cos\phi_n + x\sin\phi_n, \phi_n)$,

où $f^*(x,y)$ est la coupe tomographique estimée par la méthode de rétroprojection filtrée. Le filtre rampe $|\mu_r|$ filtre les faibles fréquences responsables du flou, mais il a l'inconvénient d'amplifier les hautes fréquences et donc de détériorer le RSB et peut rendre la reconstruction instable.

La régularisation de la reconstruction se fait en général par la multiplication du filtre rampe par une fenêtre d'apodisation (filtre passe-bas) comme Hanning, Hamming ou Butter worth. Ces différents filtres sont associés à des coupures des hautes fréquences plus au moins brusques. Le choix du filtre est un compromis entre le niveau de bruit acceptable et la résolution recherchée. L'article de Pan et al. [158] présente une analyse détaillée de la méthode FBP. La méthode de rétroprojection filtrée est la méthode de reconstruction analytique la plus utilisée en imagerie médicale. Elle présente de meilleures performances en comparaison avec les autres méthodes analytiques ; elle est simple à implanter et est rapide dans son exécution grâce à l'efficacité de l'algorithme de la transformée de Fourier rapide 1D (FFT).

4.1.3 Méthodes anlytiques 3D

Comme il a été expliqué à la section 4.1, la reconstruction tridimensionnelle à partir des acquisitions de données 3D se fait par des méthodes de reconstruction 3D approximatives. Pour ces algorithmes, les données des sinogrammes obliques sont redistribuées sur des sinogrammes droits et des algorithmes 2D sont utilisés pour reconstruire l'ensemble des coupes transversales qui formeront l'objet 3D. Cette approche produit des images acceptables seulement lorsque l'objet à imager est petit par rapport à l'ensemble du champ de vue du tomographe et que l'angle azimutal (θ) d'acceptation des données est petit (quelques degrés). La solution réside dans l'utilisation des algorithmes de reconstruction 3D qui sont plus robustes, mais plus longs à exécuter.

La méthode de référence (*Gold standard*) des algorithmes analytiques 3D est la rétroprojection filtrée 3D proposée par Kinahan et Rogers [104], qui est une extension d'une dimension de la rétroprojection filtrée 2D. Cependant, on note deux différences importantes entre les deux méthodes : la deuxième est la conséquence de la première :

1. La projection des données pour le mode d'acquisition 3D est incomplète. En effet, l'acquisition des sinogrammes obliques se fait pour des angles polaires limités (θ maximal est de l'ordre de 15°) parce que le système de détection a une forme cylindrique. De plus, pour les angles d'acquisitions $\theta = 0$, les données sont tronquées. Cette troncature augmente avec l'angle θ et implique une variation de la sensibilité en fonction de l'axe Z , comme il a été expliqué à la section 2.6.2 Pour avoir une projection des données complète en mode 3D, il faudrait que l'acquisition des données s'effectue de 0 à 180° pour les deux directions θ et ϕ. Ceci n'est possible que pour un système de détection sphérique ou cylindrique de longueur infinie. Par conséquent, la reconstruction directe 3D des données par un algorithme de rétroprojection filtrée 3D obtenue par une simple extension d'une dimension de FBP 2D est techniquement impossible.
2. Pour corriger ce problème de projection incomplète, l'algorithme de rétroprojection filtrée 3D construit d'abord les coupes transversales à partir seulement des sinogrammes droits $\theta = 0$ en utilisant la rétroprojection filtrée 2D. Ces coupes sont groupées, par la suite, pour former un estimé intermédiaire de la distribution 3D. Cette distribution 3D intermédiaire est ensuite utilisée par le problème direct (rétroprojection) pour calculer les données manquantes. Une fois que les données manquantes sont calculées, la rétroprojection filtrée 3D est similaire à celle 2D :

$$p(x_r, y_r, \phi, \theta) = \int_{-\infty}^{+\infty} f(x, y, z) dz_r, \tag{4.14}$$

où $p(x_r, y_r, \phi, \theta)$ est la projection le long de la LOR et z_r est le vecteur unitaire qui définit la direction de projection : $z_r = (\cos\phi \sin\theta, \sin\phi \cos\theta, \sin\theta)$. Les différentes étapes pour réaliser une FBP 3D sont :

1. Extraire les sinogrammes 2D droits ($\theta = 0$).
2. Reconstruire les coupes 2D transversales à partir des sinogrammes droit en utilisant la rétroprojection filtrée 2D.
3. Arranger les coupes 2D reconstruites pour former l'objet 3D.
4. Résoudre le problème direct (projection) pour calculer les données manquantes.
5. Calculer pour chaque ϕ et chaque θ la transformée de Fourier 2D de la projection correspondante :$P_2(\mu_r, \nu_r, \phi, \theta) = TF_2 p(x_r, y_r, \phi, \theta)$.
6. Filtrer dans le domaine fréquentiel chaque projection par le filtre 2D choisi $F(\mu_r, \nu_r)$: $\hat{P}(\mu_r, \nu_r, \phi, \theta) = F(\mu_r, \nu_r) P_2(\mu_r, \nu_r, \phi, \theta)$.
7. Calculer la transformation de Fourier 2D inverse de chaque projection filtrée : $\hat{p}(x_r, y_r, \phi, \theta) = TF_2^{-1} \hat{P}(\mu_r, \nu_r, \phi, \theta)$.
8. Rétroprojetter en 3D les projections 2D filtrées :

$$f^*(x, y, z) = \sum_{n=1}^{n=N} \hat{p}(x_r, \phi_n) = \sum_{n=1}^{n=N} \sum_{m=1}^{m=M} \hat{p}(X_r, y_r, \phi_n, \theta_m).$$

En comparaison à la rétroprojection filtrée 2D, la rétroprojection 3D apporte une nette amélioration du RSB grâce à l'estimation et à l'incorporation dans l'algorithme des données manquantes. Cette amélioration du RSB permet d'appliquer aux images reconstruites des filtres passe-bas ayant une haute fréquence de coupure pour améliorer la résolution. Cependant, FBP 3D nécessite un temps de calcul très long, qui n'est pas compatible avec les besoins cliniques ; 40% du temps total est consacré à l'estimation des données manquantes et leur rétroprojection [193]. Par ailleurs, la puissance des ordinateurs disponibles ne permet pas de réduire ce temps. Le développement d'autres plates-formes informatiques plus rapides entraînerait une utilisation plus large de cet algorithme.

4.1.4 Performances des méthodes analytiques

Les algorithmes analytiques 2D en général, et la rétroprojection filtrée en particulier, étaient les algorithmes les plus utilisés en clinique pour la reconstruction en TEP jusqu'à la fin des années 90. Ces algorithmes sont plus rapides par rapport aux méthodes itératives. Leur inconvénient majeur est la modélisation simplifiée du TEP qui conduit à des estimations biaisées et bruitées [193, 237]. En effet, le modèle du problème direct utilisé par ces algorithmes : 1) se base sur la projection de Radon qui considère que de la projection des données se fait perpendiculairement à la surface du cristal le long des lignes de réponse infiniment petites. Or, en TEP les cristaux ont des dimensions finies et les photons pénètrent dans le cristal non seulement par la surface de détection, mais aussi à travers les surfaces latérales, 2) néglige la variation de la sensibilité des détecteurs selon la position d'une source ponctuelle par rapport à la LOR considérée, et qui est causée principalement par la variation de l'angle solide de détection de la source, 3) ignore l'interstice (la zone morte) existant entre deux cristaux voisins, 4) ne tient pas compte des phénomènes physiques comme le parcours des positrons avant l'annihilation, la non-colinéarité de la paire de photons produite par annihilation et la diffusion dans les cristaux, et 5) considère que les données sont non bruitées et ignorent leur aspect stochastique.

Par ailleurs, malgré que les algorithmes itératifs se basent sur une modélisation plus précise du TEP, comme on va le voir par la suite, le débat n'est pas encore tranché de la supériorité des algorithmes itératifs sur les méthodes analytiques. Les deux types d'algorithmes vont continuer à coexister. Mais pour les méthodes analytiques, il y aura une migration de plus en plus des algorithmes 3D approximatifs vers les vrais algorithmes 3D.

4.2 Méthodes itératives déterministes

4.2.1 Critère d'estimation

Pour résoudre le problème inverse, les algorithmes déterministes estiment le vecteur $\mathbf{f}$ de la distribution du radiotraceur qui minimise l'erreur des moindres carrés entre les données mesurées par le tomographe et les données calculées par le modèle du problème direct (section 3.2) :

$$\mathbf{f_{MC}} = \arg\min_{f} \| \mathbf{d} - \mathbf{Af} \|^2, \tag{4.15}$$

où $\mathbf{f_{MC}}$ est le vecteur estimé de la distribution du radiotraceur et dont la longueur J est le nombre total de voxels de l'image, $\mathbf{d}$ est le vecteur d'acquisition dont la longueur I est le nombre de paires de détecteurs utilisées pour faire la collecte des données 3D en mode histogramme, et $\mathbf{A}$ est la MS qui modélise le tomographe de dimension I x J. Les données d'acquisition $\mathbf{d}$ sont supposées corrigées pour les coïncidences aléatoires et le diffusé. En annulant la dérivée de $\| \mathbf{d} - \mathbf{Af} \|^2$ par rapport $\mathbf{f}$, on obtient

$$\mathbf{f_{MC}} = \mathbf{A}^{\dagger}\mathbf{d} = (\mathbf{A^T A})^{-1}\mathbf{A^T d}, \tag{4.16}$$

où $\mathbf{A}^{\dagger}$ est la matrice inverse de $\mathbf{A}$ au sens des moindres carrés.

4.2.2 Inversion directe par décomposition en valeurs singulières

La matrice $(\mathbf{A^T A})^{-1}$ est très grande (de l'ordre de 10^7 x 10^9 éléments) et elle est très mal conditionnée (problème mal posé). Son inversion directe nécessite des supercalculateurs et un temps de calcul très long qui n'est pas acceptable en milieu clinique. De plus, l'accumulation des erreurs d'arrondissements et de troncatures amplifie énormément le bruit des données d'acquisition et rend l'estimation instable. En décomposant $\mathbf{A}$ en valeurs singulières (DVS), on aura

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{W^T}, \tag{4.17}$$

où $\mathbf{U}$ est une matrice orthogonale de dimensions I x I dont les colonnes sont les vecteurs propres de $\mathbf{A^T A}$, W est une matrice orthogonale de dimensions J x J dont les colonnes sont les vecteurs propres de $\mathbf{AA^T}$ et Σ est une matrice diagonale de dimension I x J ; $\Sigma = diag(\lambda_1, \cdots, \lambda_R)$ avec $\lambda_1 \geq \cdots > \lambda_R \geq 0$. Les valeurs λ_i $|0 \leq i \leq R$ non nulles sont appelées les valeurs singulières de $\mathbf{A}$ et elles correspondent aux racines carrées positives des valeurs propres de $\mathbf{AA^T}$. La combinaison de l'équation de la modélisation $\mathbf{d} = \mathbf{Af}$ (section 3.2) et de l'équation 4.17, permet d'écrire

$$\mathbf{d} = \mathbf{AF} = \sum_{i=1}^{i=R} \mathbf{U}_i \lambda_i (\mathbf{W}_i^T \mathbf{f}). \tag{4.18}$$

En inversant l'équation 4.18, on obtient le vecteur $\mathbf{f_{MC}}$ estimé au sens des moindres carrés

$$\mathbf{f_{MC}} = \sum_{i=1}^{i=P} \frac{(\mathbf{U}_i^T \mathbf{d})}{\lambda_i} \mathbf{W}_i, \tag{4.19}$$

où P est le rang de la matrice $\mathbf{A}$.

On note de l'équation 4.19 que la composante du bruit dans le signal **d** est beaucoup amplifiée par les grandes valeurs de $\frac{1}{\lambda_i}$. Les erreurs numériques introduites sur les valeurs $\frac{1}{\lambda_i}$ d'ordre supérieur lors du calcul de l'inverse de $(\mathbf{A^T A})^{-1}$ ou de la décomposition en DVS de **A** seront aussi amplifiées. Par conséquent, la méthode de reconstruction par moindres carrés est instable, que ce soit en calculant directement l'inverse $(\mathbf{A^T A})^{-1}$ ou en décomposant en valeurs singulières la matrice **A**. Le problème est dit mal posé et nécessite une régularisation pour atténuer la contribution des vecteurs singuliers d'ordre supérieur dans la reconstruction de l'image. Une manière de régularisation simple est d'utiliser la méthode de décomposition en valeurs singulières généralisée (SVDG). Elle consiste à tronquer dans la décomposition DVS la participation des petites valeurs singulières à partir d'un certain rang *M* inférieur au rang *P* de **A**, on aura donc

$$\mathbf{f_{MC}} = \sum_{i=1}^{i=M} \frac{(\mathbf{U}_i^T \mathbf{d})}{\lambda_i} \mathbf{W}_i, \tag{4.20}$$

où M < P. Le choix du rang de troncature doit permettre un bon compromis entre la résolution recherchée et le niveau de bruit sur l'image estimée. Le RSB s'améliore aux dépens de la résolution spatiale lorsque le rang de troncature diminue. Plusieurs critères ont été étudiés dans la littérature pour le choix du rang de la troncature [245].

L'inversion directe par la méthode de décomposition en valeurs singulières régularisées a été utilisée pour la reconstruction en mode d'acquisition 2D par Selivanov et Lecomte [203]. Cependant, à cause de la taille de la matrice **A** en mode 3D, cette méthode ne peut pas être implantée dans un contexte clinique pour le moment ; la reconstruction serait trop longue [204].

4.2.3 Méthodes algébriques

Les méthodes algébriques de reconstruction (ART : *Algebric Reconstruction Techniques*) sont des méthodes itératives minimisant le critère des moindres carrées pour résoudre les problèmes inverses et reconstruire les images à partir de leurs projections. Ces méthodes ont été proposées la première fois pour résoudre les systèmes d'équations linéaires par Kaczmarz [94]. La première publication de leur utilisation pour la reconstruction en imagerie médicale est parue en 1970 [62]. Plusieurs algorithmes ART ont été utilisés dans la reconstruction des images à partir des projections en médecine nucléaire, mais leur convergence n'est pas généralement assurée dans le cas des données bruitées. Xu et al. [247] ont présenté une forme généralisée de ces

algorithmes :

$$\hat{\mathbf{f}}^{\,\mathrm{k+1}} = \hat{\mathbf{f}}^{\,\mathrm{k}} + \mathbf{P}\dagger\,(\mathbf{d} - \mathbf{A}\hat{\mathbf{f}}^{\,\mathrm{k}}),$$
$$\mathbf{P}\dagger = r\mathbf{Q}\mathbf{A}^T, \tag{4.21}$$

où $\hat{\mathbf{f}}^{\,\mathrm{k+1}}$ est le vecteur des voxels estimé à la $(k+1)$ ième itération, $\hat{\mathbf{f}}^{\,\mathrm{k}}$ est celui estimé à k ième itération, r est un facteur de relaxation et $\mathbf{Q}$ est une matrice de régularisation. L'algorithme consiste à faire la projection de l'image $\hat{\mathbf{f}}^{\,\mathrm{k}}$, comparer cette projection avec les mesures en calculant leur différence, rétroprojeter l'erreur dans l'espace image et puis corriger le vecteur image $\hat{\mathbf{f}}^{\,\mathrm{k}}$ en lui additionnant l'erreur rétroprojetée. Si $\mathbf{Q}$ est la matrice identité, l'algorithme converge en principe vers la solution des moindres carrées. Si $\mathbf{Q} = \mathbf{Z}^{-\frac{1}{2}}$ où $\mathbf{Z}$ est la matrice de covariance des mesures $\mathbf{d}$ alors la solution va converger théoriquement vers le vecteur qui minimise le critère des moindres carrés pondérés :

$$\begin{aligned} \mathbf{f}_{\mathbf{MCP}} &= \arg\min_f (\mathbf{D} - \mathbf{A}\mathbf{f})\mathbf{Q}(\mathbf{f} - \mathbf{A}\mathbf{f}), \\ &= (\mathbf{A}^T\mathbf{Q}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{f}. \end{aligned} \tag{4.22}$$

Le critère des moindres carrés pondérés est utilisé lorsque les projections présentent des variances très différentes. Les algorithmes ART montrent une semi-convergence, c'est-à-dire que la reconstruction s'améliore durant les premières itérations, mais le rapport signal-bruit commence à se détériorer très rapidement par la suite. Leur convergence vers une solution n'est pas assurée, surtout dans le cas où il n'existe pas une solution exacte au problème inverse (mesures non consistantes) [67, 92, 178]. En fait, ces méthodes se basent sur le critère des moindres carrés qui a tendance à amplifier le bruit.

4.3 Méthodes itératives stochastiques

Les algorithmes itératifs déterministes se basent sur un modèle du système tomographe $\mathbf{d} = \mathbf{A}\mathbf{f}$ qui considère que les mesures $\mathbf{d}$ sont des valeurs déterministes et que le vecteur image à estimer $\mathbf{f}$ est aussi déterministe. La convergence de ces algorithmes n'est pas assurée en présence de bruit. Un modèle plus réaliste consiste à considérer la nature probabiliste de l'émission d'un

positron et celui du bruit qui entache les mesures. Donc l'équation du problème direct devient :

$$\bar{\mathbf{d}} = E(\mathbf{D}) = \mathbf{A}\mathbf{f}, \tag{4.23}$$

où le vecteur des mesures $\mathbf{D}$ est une variable aléatoire de valeur moyenne $\mathbf{d}$. Les coefficients $\mathbf{A}_{i,j}$ de la matrice système $\mathbf{A}$ désignent les probabilités de détecter une coïncidence qui a eu lieu au voxel j par la paire de détecteurs i.

4.3.1 Estimateurs statistiques au sens du maximum de vraisemblance (MV)

4.3.1.1 Critère d'estimation au sens du MV

Les estimateurs au sens du maximum de vraisemblance cherchent la solution $\mathbf{f_{MV}}$ qui maximise la probabilité pour que les mesures $\mathbf{d} = [d_1, d_2, \cdots, d_I]^T$ de la variable aléatoire $\mathbf{D}$ se produisent :

$$\mathbf{f_{MV}} = \arg\max_f \mathbf{P_{D=d|f}}, \tag{4.24}$$

où $\mathbf{f_{MV}}$ est la distribution du traceur estimée au sens du MV et $\mathbf{P_D} = \mathbf{d}|\mathbf{f}$, appelée fonction objective au sens du MV, est la probabilité conditionnelle pour que la variable aléatoire $\mathbf{D}$ prenne les mesures $\mathbf{d}$ considérant le vecteur image $\mathbf{f}$. Donc, l'estimateur MV a pour objectif de maximiser la fidélité de la reconstruction aux mesures. Par conséquent, cet estimateur est non biaisé, autrement dit l'espérance $E[\mathbf{f_{MV}}]$ tend vers la vraie solution $\mathbf{f}$ lorsque le nombre de coïncidences acquises est grand [178]. Par contre, si les données mesurées sont bruitées, l'image estimée sera bruitée aussi. Un filtrage basse fréquence est alors nécessaire pour réduire le niveau de bruit aux dépens de l'exactitude de la reconstruction. Par ailleurs, le critère MV est le critère qui permet d'obtenir des solutions moins bruitées (faible variance) en comparaison aux autres critères non biaisés tels que le critère au sens des moindres carrés [178]. Si on considère que la variable aléatoire D suit la loi de Poisson alors on aura

$$\mathbf{P_{D=d|f}} = \prod_{i=1}^{I} \frac{\bar{d}_i^d}{d_i!} exp(-\bar{d}_i). \tag{4.25}$$

L'image estimée $\mathbf{f_{MV}}$ au sens du MV sera obtenue en maximisant la fonction 4.25. Par ailleurs, puisque le logarithme est une fonction monotone, il est plus facile et plus judicieux de maximiser le logarithme de cette fonction pour obtenir la solution. En appliquant le logarithme à la fonction $\mathbf{P_{D=d|f}}$ pour chercher la solution, on obtient la fonction $\mathbf{L_{D=d|f}}$ appelée dans

la littérature fonction de vraisemblance :

$$\begin{aligned} \mathbf{L}_{\mathbf{D}=\mathbf{d}|\mathbf{f}} &= \ln(\mathbf{P}_{\mathbf{D}=\mathbf{d}|\mathbf{f}}), \\ &= \sum_{i=1}^{i=I} d_i \ln(\bar{d}_i) - \bar{d}_i - \ln(d_i!). \end{aligned} \tag{4.26}$$

En combinant les équations 4.26 et 4.24, on obtient :

$$\begin{aligned} \mathbf{f}_{\mathbf{MV}} &= \arg\max_f \sum_{i=1}^{i=I} d_i \ln(\bar{d}_i) - \bar{d}_i - \ln(d_i!), \\ &= \arg\max_f \sum_{i=1}^{i=I} d_i \ln((\mathbf{Af})_i) - (\mathbf{Af})_i - \ln(d_i!). \end{aligned} \tag{4.27}$$

La solution estimée au sens du MV, sera donc obtenue en maximisant la fonction de vraisemblance $\mathbf{L}_{\mathbf{D}=\mathbf{d}|\mathbf{f}}$.

4.3.1.2 L'algorithme de maximisation de vraisemblance MLEM

Les algorithmes les plus utilisés pour maximiser la fonction de coût au sens du MV $\mathbf{L}_{\mathbf{D}=\mathbf{d}|\mathbf{f}}$ pour la reconstruction des images en médecine nucléaire sont des algorithmes itératifs de type gradient [193] :

$$\hat{\mathbf{f}}_j^{\mathrm{k+1}} = \hat{\mathbf{f}}_j^{\mathrm{k}} + \lambda_j^k \frac{\partial \mathbf{L}_{\mathbf{D}=\mathbf{d}|\hat{\mathbf{f}}^{\mathrm{k}}}}{\partial \hat{\mathbf{f}}_j^{\mathrm{k}}}, \tag{4.28}$$

où λ_j^k est le pas de correction. En choisisant $\lambda_j^k = \frac{\hat{f}_j^k}{\sum_{n=1}^{N} \mathbf{A}_{i,j}}$, on obtient l'algorithme itératif MLEM proposé par Shepp et Vardi [205] :

$$\hat{\mathbf{f}}_j^{\mathrm{k+1}} = \frac{\hat{\mathbf{f}}_j^{\mathrm{k}}}{\sum_{i=1}^{i=I} \mathbf{A}_{i,j}} \sum_{i=1}^{I} \mathbf{A}_{i,j} \frac{\mathbf{d}_i}{\sum_{n=1}^{J} \mathbf{A}_{i,n} \hat{\mathbf{f}}_n^{\mathrm{k}}}. \tag{4.29}$$

Sous forme matricielle, on aura :

$$\hat{\mathbf{f}}^{\mathrm{k+1}} = \frac{\hat{\mathbf{f}}^{\mathrm{k}}}{\mathbf{A}^T \mathbf{1}} \mathbf{A}^T \frac{\mathbf{d}}{\mathbf{A}\hat{\mathbf{f}}^{\mathrm{k}}}, \tag{4.30}$$

où $\mathbf{1}$ est le vecteur dont tous les éléments ont la valeur 1. L'algorithme MLEM a été décrit la première fois par Dempster et al. [52] pour résoudre tous les problèmes d'estimation au sens du MV. La version décrite ci-dessus (3.30) pour la reconstruction des images en médecine nu-

cléaire a été proposée par Shepp et Vardi [205]. Cet algorithme est très simple à implémenter numériquement. En effet, il consiste à : 1) projeter l'image actuelle $\hat{\mathbf{f}}^{\,\mathrm{k}}$ dans l'espace des projections : $\mathbf{P} = \mathbf{A}\hat{\mathbf{f}}^{\,\mathrm{k}}$, 2) calculer le rapport des données mésurées $\mathbf{d}$ et les valeurs calculées par projection $\mathbf{P}$ pour déterminer le facteur de correction : $\mathbf{F_{cor}} = \frac{\mathbf{d}}{\mathbf{P}}$, 3) rétroprojeter ce rapport dans l'espace image pour calculer la matrice gradient : $\mathbf{G} = \mathbf{A}^T\mathbf{F_{cor}}$, 4) actualiser l'image $\hat{\mathbf{f}}^{\,\mathrm{k}}$ en la multipliant par le gradient : $\hat{\mathbf{f}}^{\,\mathrm{k+1}} = \hat{\mathbf{f}}^{\,\mathrm{k}}\mathbf{G}$, et 5) normaliser l'image actualisée : $\hat{\mathbf{f}}^{\,\mathrm{k+1}} = \frac{\hat{\mathbf{f}}^{\,\mathrm{k+1}}}{\mathbf{A}^T\mathbf{1}}$. Par rapport aux algorithmes ART, la convergence de MLEM est prédictible et consistante car l'erreur calculée et le facteur de correction sont des termes multiplicatifs alors qu'ils sont des termes additifs pour les algorithmes ART [238]. En outre, MLEM présente deux inconvénients majeurs : le premier est que sa convergence est très lente (50 à 100 itérations) et le deuxième est que les images reconstruites sont bruitées, car le critère de MV, sur lequel se base l'algorithme, conduit à des estimés bruités, mais qui ne sont pas biaisés. Donc plus le nombre d'itérations augmente plus la résolution de l'image s'améliore, mais plus le bruit augmente. Des critères pour stopper les itérations ont été développés [68, 121, 210].

4.3.1.3 L'algorithme de maximisation de vraisemblance OSEM

Hudson et Larkin [84] ont proposé une variante rapide de l'algorithme MLEM appelé algorithme OSEM. Elle consiste à : 1) diviser le vecteur de mesure $\mathbf{d}$ en un nombre p de sous ensembles $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_p$ qui forment une partition ; p est l'ordre de la partition et 2) estimer l'image en appliquant l'algorithme MLEM sur les sous vecteurs $\mathbf{S}_q$, $q = 1 \cdots p$ successivement :

$$\hat{\mathbf{f}}_j^{k+1} = \frac{\hat{\mathbf{f}}_j^k}{\sum_{i \in S_q} \mathbf{A}_{i,j}} \sum_{i \in S_q} \mathbf{A}_{i,j} \frac{\mathbf{d}_i}{\sum_{n=1}^{J} \mathbf{A}_{i,n} \hat{\mathbf{f}}_n^k}. \tag{4.31}$$

L'actualisation de l'espérance $\mathbf{f}_j^k$ se fait successivement à partir des données $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_p$. Chaque étape de l'actualisation est appelée une sous-itération et un ensemble de p sous-itérations successives forme une itération. Le temps d'exécution d'une itération OSEM est comparable à celui d'une itération MLEM. Cependant, l'algorithme OSEM converge p fois plus rapidement pour un ordre p que l'algorithme MLEM [84]. Généralement la convergence se fait lors des premières itérations, ce qui permet de diminuer le temps de calcul. Pour cette raison OSEM est l'algorithme le plus utilisé actuellement dans la reconstruction des images en TEP que ce soit en clinique ou en recherche [178].

L'inconvénient de l'OSEM est que la convergence vers la solution MV est biaisée, les images sont bruitées et la solution oscille entre des limites à chaque itération. Plus on augmente l'ordre p de la partition, plus on accélère le calcul, mais plus l'image estimée est bruitée et la convergence vers une seule solution est compromise. Le choix de p est un facteur important pour un compromis entre la rapidité de calcul et la convergence vers la solution.

Des solutions ont été proposées pour remédier à la perte de convergence de l'algorithme OSEM, entre autres les algorithmes types *relaxed OS method* qui font diminuer le pas de correction à chaque itération. Parmi ces algorithmes, on trouve l'algorithme RAMLA (*Row-Action Maximum-Likelihood*) proposé par Browne et De Pierro [31] :

$$\hat{\mathbf{f}}_j^{k+1} = \hat{\mathbf{f}}_j^k + \lambda^k \frac{\hat{\mathbf{f}}_j^k}{\sum_{i\in S_q} \mathbf{A}_{i,j}} \left(\sum_{i\in S_q} \mathbf{A}_{i,j} \frac{\mathbf{d}_i}{\sum_{n=1}^{J} \mathbf{A}_{i,n} \hat{\mathbf{f}}_n^k} - \sum_{i\in S_q} \mathbf{A}_{i,j} \right). \tag{4.32}$$

où λ^k est le facteur de relaxation. Le choix de ce facteur est aussi un compromis entre la rapidité de la convergence de l'algorithme et la convergence vers la solution MV ; une sur-relaxation (décroissance rapide de λ^k) conduit à une solution différente de MV et une sous-relaxation entraîne une augmentation importante du nombre d'itérations nécessaires à la convergence.

Une autre approche pour réduire le temps d'exécution des algorithmes MLEM et OSEM a été explorée, il consiste à paralléliser ces algorithmes pour les exécuter sur des plates-formes de calcul parallèle comme des grappes de calcul [21, 40, 71, 119] ou bien sur des GPUs [69, 70, 101, 198, 256].

4.3.2 Estimateurs au sens du maximum *a posteriori* (MAP)

4.3.2.1 Critère d'estimation au sens du MAP

Le critère au sens du MV se fie exclusivement aux mesures pour estimer la solution qui est considérée comme la plus probable. L'image obtenue est généralement bruitée et risque même d'être biaisée si les mesures sont incomplètes comme dans le cas des acquisitions TEP. Le critère d'estimation au sens du maximum a posteriori (MAP) considère que la solution est aussi probabiliste et essaie de régulariser la solution MV en imposant un *a priori* sur celle-ci. Le critère MAP est formulé par

$$\mathbf{f_{MAP}} = \arg\max_f P_{\mathbf{F}=\mathbf{f}|\mathbf{D}=\mathbf{d}}. \tag{4.33}$$

$\mathbf{f_{MAP}}$ est donc la distribution du traceur qui maximise la densité de probabilité de la variable aléatoire $\mathbf{F}$ sachant que les mesures $\mathbf{d}$ de la variable aléatoire $\mathbf{D}$ ont été réalisées. La règle de Bayes permet d'écrire

$$P_{\mathbf{F}|\mathbf{D}} = \frac{P_{\mathbf{D}|\mathbf{F}} P_{\mathbf{F}}}{P_{\mathbf{D}}}. \tag{4.34}$$

Donc :

$$\mathbf{f_{MAP}} = \arg\max_f \frac{P_{\mathbf{D}|\mathbf{F}} P_{\mathbf{F}}}{P_{\mathbf{D}}} = \arg\max_f P_{\mathbf{D}|\mathbf{F}} P_{\mathbf{F}}. \tag{4.35}$$

L'estimation de la solution se fait à partir de la combinaison de l'information fournie par les mesures $P_{\mathbf{D}|\mathbf{F}}$ et l'information *a priori* $P_{\mathbf{F}}$ disponible sur la variable aléatoire $\mathbf{F}$. En considérant le logarithme de cette dernière équation, on obtient

$$\begin{aligned} \mathbf{f_{MAP}} &= \arg\max_f (\log(P_{\mathbf{D}|\mathbf{F}}) + \log(P_{\mathbf{F}})), \\ &= \arg\max_f (L_{\mathbf{D}|\mathbf{F}} + \log(P_{\mathbf{F}})). \end{aligned} \tag{4.36}$$

On note donc que le critère MAP est composé de deux termes ; le premier $L_{D|F}$ est tout simplement le critère MV et le deuxième $\log(P_{\mathbf{F}})$ est un terme régulateur qui permet de pénaliser les solutions inappropriées. Le MAP est équivalent donc au MV régularisé. La plus grande difficulté pour le critère MAP est la détermination de l'*a priori* $P_{\mathbf{F}}$. Des auteurs ont utilisé des images anatomiques obtenues par tomodensitométrie ou par résonance magnétique comme *a priori* [10, 17, 42, 157, 206, 211]. Cependant, cette approche produit parfois des images biaisées, car la nature de l'information provenant de deux modalités d'imagerie est différente. Par exemple, des contours obtenus par résonance magnétique peuvent ne correspondre à aucun contour de l'imagerie fonctionnelle TEP. L'approche usuelle est l'utilisation des *a priori* génériques qui auront tendance à lisser les images pour pénaliser le bruit et réduire leur variance.

Pour chercher les propriétés locales des structures, plusieurs auteurs ont utilisé pour *a priori* les densités de Gibbs. En effet, les propriétés markoviennes de ces distributions et leur formalisme mathématique simple font d'elles un outil attrayant pour décrire les propriétés locales des images. Si on note par $S = 1, 2, \cdots, J$ l'ensemble des indices des voxels de l'image $\mathbf{f}$ à estimer, un système de voisinage $v = (v_s,\ s \in S)$ dans un champ markovien est formé des parties v_s de S

qui vérifient les propriétés suivantes :

$$s \notin \nu_s,$$
$$s \in \nu_t \Leftrightarrow t \in \nu_s. \tag{4.37}$$

Autrement dit, chaque voxel s n'appartient pas à son voisinage ν_s et un voxel s est un voisin du voxel t si et seulement si t est un voisin de s. On appelle clique relative au système de voisinage ν, un sous ensemble q de S où deux voxels de q sont voisins au sens de ν. La forme générale des distributions de Gibbs est

$$P_{\mathbf{F}=\mathbf{f}} = \frac{1}{Z} exp(-\beta U(\mathbf{f})), \tag{4.38}$$

où Z est la constante de normalisation de la probabilité, β est un paramètre de pondération qui module la tangente de la distribution autour des maxima et $U(\mathbf{f})$ est la fonction de l'énergie de Gibbs. Cette fonction est la somme des fonctions potentielles sur l'ensemble des cliques :

$$U(\mathbf{f}) = \sum_{q \in Q} V_q(\mathbf{f}), \tag{4.39}$$

où Q est l'ensemble des cliques. L'énergie $U(\mathbf{f})$ est généralement définie sur des cliques d'ordre 2 comme suit :

$$U(\mathbf{f}) = \sum_{i=1}^{i=J} \sum_{j>i, j \in q_j} V_{i,j}(\mathbf{f}_i - \mathbf{f}_j). \tag{4.40}$$

Les fonctions potentielles $V_{i,j}(\mathbf{f}_i - \mathbf{f}_j)$ ont la propriété d'augmenter en fonction de la différence de la densité de distribution entre les voxels voisins i et j. Par conséquent, l'énergie $U(\mathbf{f})$ augmente pour les images présentant une grande variation d'intensité entre les voxels voisins ce qui diminue la probabilité de réalisation *a priori* de ces images. L'équation 4.36 devient donc

$$\mathbf{f_{MAP}} = \arg\max_{f}(L_{\mathbf{D}|\mathbf{F}} - \beta U(\mathbf{f})). \tag{4.41}$$

On note que le terme d'*a priori* $\beta U(\mathbf{f})$ permet la régularisation de la solution au sens du maximum de vraisemblance. C'est un terme important qui assure un compromis entre la résolution recherchée (degré de lissage) et le rapport signal-bruit accepté.

Plusieurs fonctions potentielles $V_{i,j}(\mathbf{f}_i - \mathbf{f}_j)$ ont été utilisées dans la littérature pour permettre un lissage de la distribution du traceur à l'intérieur d'un organe ou sur une masse de cellules

ayant les mêmes caractéristiques métaboliques toute en préservant les changements brusques de la distribution entre les frontières de ces différents tissus [10, 64, 68, 108, 140, 141]. La plupart de ces fonctions sont des fonctions croissantes de la variation d'intensité $|\mathbf{f}_i - \mathbf{f}_j|$. En prenant $V_{i,j}(\mathbf{f}_i - \mathbf{f}_j) = (\mathbf{f}_i - \mathbf{f}_j)^2$, on obtient l' *a priori* de Gauss-Markov qui a tendance à pénaliser des images ayant des contours avec des variations brusques. Pour augmenter la probabilité de détecter ces contours, Bouman et Sauer [28] ont proposé l'utilisation du modèle p-gaussien $V_{i,j}(\mathbf{f}_i - \mathbf{f}_j) = (\mathbf{f}_i - \mathbf{f}_j)^p$ où $1 < p < 2$.

Le superparamètre β est un facteur de pondération entre la fidélité de l'estimé aux mesures et la contrainte exprimée par la fonction d'énergie $U(\mathbf{f})$. Un facteur β nul permet d'obtenir la solution au sens du maximum de vraisemblance qui est une solution non biaisée mais bruitée (variance élevée). En augmentant β, on donne plus de poids à la fonction énergie $U(\mathbf{f})$. La détermination du paramètre β se fait, généralement, manuellement en se basant sur l'expérience de l'utilisateur pour obtenir la solution voulue. Par ailleurs, des méthodes automatiques ont été étudiées dans la littérature pour calculer β [73, 140, 257].

4.3.2.2 Algorithmes de maximisation du critère MAP

Puisque les deux critères d'estimation MV et MAP sont similaires, la plupart des algorithmes de maximisation du critère MAP ont été obtenus en modifiant ceux qui sont développés pour le critère MV pour y inclure le terme d'*a priori*, entre autres : les algorithmes EM ont été notamment généralisés pour maximiser le critère MAP (GEM : *Generalized Expectation Method*).

L'algorithme MAP-EM qui est l'analogue de MLEM pour MAP a la forme suivante [64] :

$$\hat{\mathbf{f}}_j^{\,\mathrm{k+1}} = \frac{\hat{\mathbf{f}}_j^{\,\mathrm{k}}}{\sum_{i=1}^{i=I} \mathbf{A}_{i,j} + \beta \frac{\partial U(\mathbf{f})}{\partial(\mathbf{f})|\mathbf{f}=\hat{\mathbf{f}}^{\,\mathrm{k+1}}}} \sum_{i=1}^{I} \mathbf{A}_{i,j} \frac{\mathbf{d}_i}{\sum_{n=1}^{J} \mathbf{A}_{i,n} \hat{\mathbf{f}}_n^{\,\mathrm{k}}}. \tag{4.42}$$

Cependant, le terme $\frac{\partial U(\mathbf{f})}{\partial(\mathbf{f})|\mathbf{f}=\hat{\mathbf{f}}^{\,\mathrm{k+1}}}$ au dénominateur cause un problème pratique d'évaluation puisque l'image $\hat{\mathbf{f}}^{\,\mathrm{k+1}}$ estimée n'est pas disponible à l'instant k pour évaluer le prochain terme de la série d'itérations. Ainsi Green [64] a proposé un algorithme appelé MAP-EM OSL (OSL : *One Step Late*) qui utilise l'image antérieure $\hat{\mathbf{f}}^k$ pour estimer cette dérivée :

$$\hat{\mathbf{f}}_j^{\,\mathrm{k+1}} = \frac{\hat{\mathbf{f}}_j^{\,\mathrm{k}}}{\sum_{i=1}^{i=I} \mathbf{A}_{i,j} + \beta \frac{\partial U(\mathbf{f})}{\partial(\mathbf{f})|\mathbf{f}=\hat{\mathbf{f}}^{\,\mathrm{k}}}} \sum_{i=1}^{I} \mathbf{A}_{i,j} \frac{\mathbf{d}_i}{\sum_{n=1}^{J} \mathbf{A}_{i,n} \hat{\mathbf{f}}_n^{\,\mathrm{k}}}. \tag{4.43}$$

Lange [107] a démontré que l'algorithme MAP-EM OSL ne converge vers la solution MAP que pour certaines formes d'*a priori* et il a proposé une version améliorée de cet algorithme pour améliorer ses propriétés de convergence. Par ailleurs, d'autres types d'algorithmes ont été développés pour améliorer la convergence des algorithmes GEM vers la solution MAP. Entre autres : les algorithmes connus sous le nom SAGE (*Space Alternating Generalized EM algorithm*) [58, 129]. Ces derniers se basent sur une autre approche qui consiste à corriger les voxels séquentiellement, au lieu de simultanément comme dans le cas des algorithmes GEM pour contrôler la convergence vers la solution MAP à chaque mise à jour d'un voxel.

4.3.3 Reconstructions stochastiques à partir des données en mode liste

Les méthodes de reconstruction à partir des données en mode histogramme (sinogrammes) décrit ci-dessus sont les méthodes les plus utilisées en clinique. En effet, le mode histogramme permet d'accélérer la reconstruction en utilisant deux approches. La première consiste à redistribuer les données d'acquisition 3D sur des sinogrammes 2D (section 4.1), et la deuxième compresse ces données en combinant plusieurs sinogrammes voisins pour réduire leur nombre (section 2.6.3). Mais il a été démontré que ces deux approches affectent la précision de la quantification surtout dans la périphérie de la région d'intérêt [241].

Par ailleurs, les méthodes de reconstruction stochastiques à partir des données en mode liste commencent à susciter de plus en plus d'intérêt en clinique. En effet, pour les systèmes modernes d'acquisition ayant un grand nombre de lignes de réponses (10^5 à 10^9), la reconstruction à partir de ce mode est plus rapide que la reconstruction à partir des sinogrammes pour les courtes acquisitions où le nombre d'événements est inférieur au nombre des lignes de réponses, comme dans le cas des acquisitions dynamiques. D'autre part, le mode liste permet d'implenter facilement la correction du mouvement et le temps de vol pour améliorer la précision de la quantification. Ce mode permet aussi d'effectuer la reconstruction 4D en utilisant des fonctions de compositions spatio-temporelles.

Cependant, pour les longues acquisitions, le temps de calcul des algorithmes stochastiques de reconstruction à partir du mode liste est encore trops long pour que ces derniers soient utilisés en clinique. Pour contourner cet obstacle et introduire la reconstruction à partir du mode liste en clinique, plusieurs travaux ont été effectués cette dernière décennie pour exécuter ces méthodes sur des plates-formes de calcul parallèles, entre autres sur les grappes de processeurs et sur les cartes graphiques [45, 170–173, 189, 197, 199]. Les travaux d'accélération ont porté plus

spécifiquement sur les deux algorithmes itératifs stochastiques LM-EM et LM-OSEM.

4.3.3.1 Les algorithmes mode liste LM-EM et LM-OSEM

L'algorithme LM-EM a été déduit par Barrett et al. [20] et Reader et al. [190] de l'algorithme de reconstruction à partir des sinogrammes MLEM (section 4.3.1.2) :

$$\begin{aligned} \hat{\mathbf{f}}_j^{\text{m}} &= \frac{\hat{\mathbf{f}}_j^{\text{m-1}}}{\mathbf{N}_j} \sum_{k=1}^{M} \mathbf{A}_{i_k,j} \frac{1}{\sum_{j=1}^{J} \mathbf{A}_{i_k,j} \hat{\mathbf{f}}_j^{\text{m-1}} + \frac{s_{i_k}+r_{i_k}}{a_{i_k}\varepsilon_{i_k}}}, \\ \mathbf{N}_j &= \sum_{i=1}^{I} \mathbf{A}_{i,j} \mathbf{a}_i \varepsilon_i, \end{aligned} \tag{4.44}$$

où M est le nombre d'événements mesurés, I le nombre total de lignes de réponses, J le nombre de voxels, $\mathbf{A}_{i_k,j}$ est le coefficient de la MS qui correspond au voxel j et à la LOR associée à la paire de détection i_k ayant détecté l'événement k, $\mathbf{N}_j$ est le coefficient j de la matrice de sensibilité qui définit la probabilité qu'un événement de coïncidence ayant lieu dans le voxel j soit détecté par le système d'acquisition, a_i est le facteur de correction de l'atténuation calculé par la projection le long des LORs de la matrice 3D des coefficients d'atténuation estimée à partir d'un examen de tomodensitométrie, ε_i est le facteur qui caractérise la sensibilité de détection de la paire des détecteurs associée à la LOR associée à l'événement i, et s_{i_k} et r_{i_k} sont respectivement le nombre moyen des événements diffusés et celui des événements aléatoires détectés sur la LOR i_k. Les coefficients $\mathbf{N}_j$ et a_i sont calculés en considérant toutes les LORs possibles qui traversent le patient.

La fonction de vraisemblance L_{List} pour le mode liste est déduite de la fonction de vraisemblance $\mathbf{L}_{\mathbf{D}=\mathbf{d}|\mathbf{f}}$ défini pour le mode histogramme dans l'équation 4.26 [85] :

$$\begin{aligned} \mathbf{L}_{List} &= \ln(\mathbf{P}_{\mathbf{D}=\mathbf{d}|\mathbf{f}}), \\ &= \sum_{i=1}^{M} \ln(\mathbf{A}\mathbf{f})_i - \sum_{j=1}^{J} (\mathbf{S}_j \mathbf{f}_j). \end{aligned} \tag{4.45}$$

Comme l'algorithme MLEM, LM-EM nécessite un grand nombre d'itérations pour converger vers la solution MV. Donc pour accélérer la convergence, l'algorithme LM-OSEM a été introduit en s'inspirant de l'algorithme OSEM. LM-OSEM consiste à : 1) diviser l'espace des événements en mode liste S en un nombre l de sous ensembles d'événements $\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_p$ qui forment une

partition ; p est l'ordre de la partition, et 2) actualiser l'image par l'algorithme LM-EM en utilisant successivement les sous-ensembles d'événements $\mathbf{L}_i$, $\mathrm{i} = 1 \cdots p$. Chaque actualisation est une sous-itération et un ensemble de l sous-itérations successives forme une itération. Mathématiquement, l'algorithme LM-OSEM est formulé par

$$\hat{\mathbf{f}}_j^{\mathrm{m}} = \frac{\hat{\mathbf{f}}_j^{\mathrm{m-1}}}{\mathbf{N}_j} \sum_{k \in L_q} \mathbf{A}_{i_k,j} \frac{1}{\sum_{j=1}^{J} \mathbf{A}_{i_k,j} \hat{\mathbf{f}}_j^{\mathrm{m-1}} + \frac{s_{i_k} + r_{i_k}}{a_{i_k} \epsilon_{i_k}}}. \tag{4.46}$$

$\hat{\mathbf{f}}_j^{\mathrm{m}}$ est la valeur du voxel j estimée après m sous-itérations et L_q est le sous-ensemble d'événements de la partition pour $q = m \bmod p$. Les autres termes de l'équation sont définis au même titre que pour l'équation 4.44 de LM-EM. L'algorithme LM-OSEM se comporte comme l'algorithme OSEM, il converge plus rapidement que LM-EM, mais la solution est biaisée et elle est trop bruitée.

4.4 Reconstruction 4D

Deux approches ont été utilisées pour la reconstruction de la distribution du traceur en 4D. La première consiste à réaliser une acquisition dynamique en divisant la durée totale d'examen en une succession d'intervalles de temps d'acquisition (fenêtres temporelles), typiquement une trentaine de trames, puis à reconstruire les images dynamiques les unes indépendamment des autres à partir du modèle :

$$\mathbf{d}_j = \mathbf{A}_j \mathbf{f}_j, \tag{4.47}$$

où $\mathbf{d}_j, \mathbf{f}_j$ et $\mathbf{A}_j$ sont respectivement le vecteur de données, l'image à estimer et la MS relatifs au $j^{\text{ième}}$ intervalle temporel d'acquisition $[T_{j-1}, T_{j+1}]$. Pour un système stationnaire, les matrices systèmes sont invariantes dans le temps $A_1 = A_2 = ... = A_J = A$. Cette approche se caractérise par une faible résolution temporelle. En effet les intervalles d'acquisitions $[T_{j-1} T_{j+1}]$ sont assez larges pour permettre une bonne statistique et donc un bon RSB. Dans le cas des études dynamiques pour l'évaluation de la fonctionnalité des organes tels que le cœur, les reins et les poumons, le facteur RSB est plus important que la résolution temporelle, donc cette approche est valable. Cependant, les paramètres cinétiques estimés pour les modèles compartimentaux pharmacocinétiques à partir des images dynamiques reconstruites sont généralement biaisés en raison de la mauvaise résolution temporelle.

La deuxième approche consiste en une reconstruction des images 4D à partir des données en mode liste en utilisant des fonctions de décompositions spatio-temporelles 4D [63, 117, 118, 192, 229, 230, 239]. Quoique que cette dernière approche permette une reconstruction 4D avec une bonne résolution temporelle, son utilisation en clinique est très limitée. En effet, puisque la reconstruction se fait à partir du mode liste qui ne permet pas une compression de données, le temps de calcul est trop long pour pouvoir utiliser en clinique de routine cette approche de reconstruction 4D.

Par ailleurs, les travaux effectués ces dernières années pour accélérer la reconstruction 3D à partir des données en mode liste sur des plates-formes de calcul parallèle peu onéreuses comme les GPU sont prometteuses. Ces travaux permettent de conclure que la réduction du temps de reconstruction 4D utilisant les fonctions de décompositions spatio-temporelles 4D à un niveau acceptable cliniquement est possible surtout avec le développement des cartes GPU de plus en plus puissantes.

4.5 Quantification

L'indice de quantification des images TEP le plus largement utilisé en clinique est le SUV (*Standardized Uptake Value*). Il est défini par le rapport entre la fixation du traceur et la dilution homogène du traceur dans le volume du patient :

$$\text{SUV} = \frac{\text{Fixation (kBq/mL)}}{\text{dose injectée (kBq) / poids du patient (g)}}, \tag{4.48}$$

où la fixation en kBq/mL est obtenue en multipliant les images reconstruites en événements/voxel par un facteur d'étalonnage du tomographe. L'étalonnage est généralement effectué par l'acquisition des données sur un cylindre dont la concentration radioactive est connue. Le facteur d'étalonnage est alors calculé en comparant le nombre d'événements par voxel à la concentration radioactive (kBq/mL). Le terme du dénominateur correspond à l'activité injectée au patient au moment de l'acquisition normalisée par son poids. Une valeur de 10 pour le SUV dans une lésion signifie que la fixation du traceur dans la lésion est 10 fois supérieure à la dilution uniforme du traceur. Cependant, plusieurs études ont montré que SUV dépend énormément du protocole d'acquisition et du traitement des données, des paramètres physiologiques du patient tels que le poids, la masse maigre et le taux de glycémie [2, 26, 27, 33, 240]. De plus, le SUV inclut le traceur non métabolisé se trouvant dans le sang vasculaire, dans l'espace extracellulaire et à

l'intérieure des cellules [59].

Ainsi, des algorithmes ont été développés pour extraire des paramètres physiologiques importants pour la clinique tels que le taux de perfusion, le taux de métabolisme cellulaire du traceur et la densité des récepteurs. Ils se basent sur les modèles compartimentaux pour décrire la cinétique du traceur (figure 4.5). Chaque compartiment d'un modèle définit un état possible du traceur, spécialement sa localisation physique (le milieu vasculaire, milieu extracellulaire ou intracellulaire) et son état biochimique (comme son métabolisme cellulaire, sa liaison avec des récepteurs, etc.). L'échange dynamique du traceur entre les différents compartiments est décrit par des équations différentielles du premier ordre qui lient les concentrations et dont les coefficients sont les constantes cinétiques d'échange entre les compartiments. La détermination des constantes, ou des combinaisons algébriques des constantes permet la quantification des paramètres physiologiques recherchés. L'objectif de la modélisation est donc de déterminer les constantes cinétiques à partir des acquisitions. La détermination des paramètres physiologiques se fait pour des régions d'intérêt, mais aussi pour chaque voxel pour générer des images paramétriques.

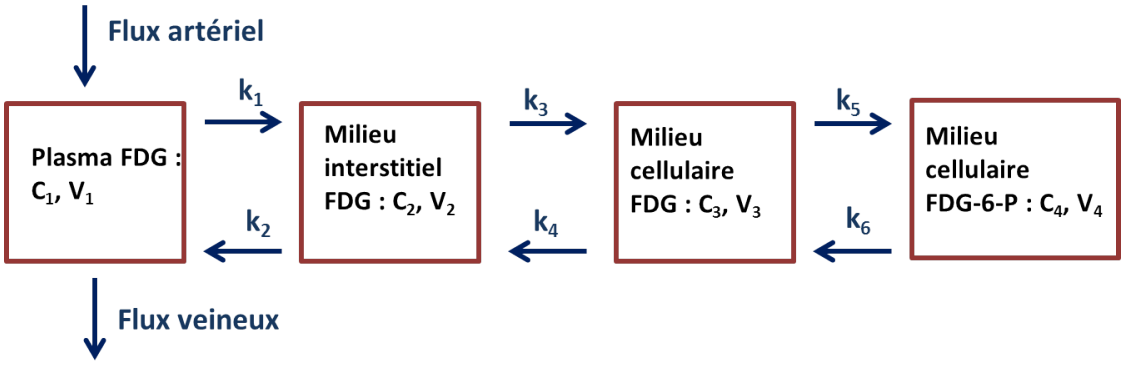


Figure 4.5 – Modèle 4 compartiments pour le FDG : K_1 (mL min^{-1} g^{-1}) , K_2, K_3, k_4, K_5 et k_6 (min^{-1}) sont les constantes cinétiques d'échange à estimer, C_1,C_2, C_3 et C_4 sont les concentrations du traceur en mL/g, et V_1, V_2, V_3 et V_4 sont les volumes de distribution en mL.

La méthode usuelle pour la détermination des constantes cinétiques est la reconstruction des images dynamiques, puis la reconstruction des courbes temps-activité pour les régions d'intérêt à partir des images reconstruites et enfin l'estimation des paramètres cinétiques en utilisant les méthodes d'analyse graphique. Les méthodes d'analyse graphique les plus utilisées sont la méthode de Patlak [163], la méthode de Logan [122] et la méthode SKM (*Simplified kinetic method*) développée par Hunter et al. [86]. Ces méthodes se basent sur des transformations mathématiques

des données mesurées pour obtenir une relation linéaire dont la pente est le paramètre cinétique ou physiologique à estimer. Les images paramétriques sont obtenues en appliquant cette approche sur les voxels pour déterminer les constantes cinétiques associées à chaque voxel.

Cependant, cette approche indirecte pour déterminer les images paramétriques à partir des images dynamiques souffre d'un manque d'exactitude et d'un faible RSB. Ces problèmes sont dus à une mauvaise résolution temporelle des images dynamiques et un mauvais RSB de ces images [191, 233] (figure 4.6). Pour améliorer la précision des images paramétriques reconstruites, une autre approche suscite de plus en plus d'intérêt ces dernières années. Elle consiste à construire ces images directement à partir des projections ou des données en mode liste [98, 128, 132, 191, 224, 233, 249]. Mais comme dans le cas de la reconstruction 4D, la reconstruction directe des images paramétriques à partir des données brutes n'est pas encore une technique utilisée dans la clinique de routine à cause du temps de calcul qui est encore long par rapport au besoin clinique.

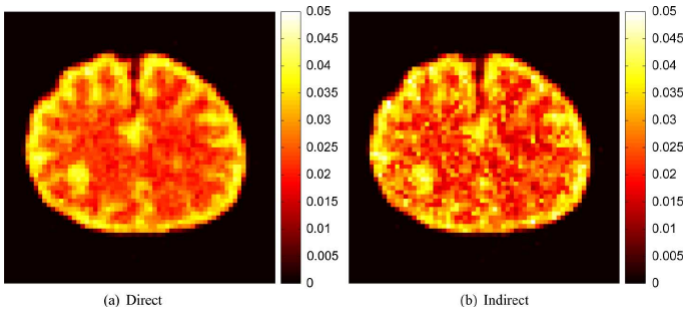


Figure 4.6 – Comparaison entre la constante cinétique k_1 estimée directement à partir des données d'acquisition brutes (sinogrammes) et indirectement à partir des images reconstruites. Images tirées de Wang et Qi [233].

4.6 Conclusion et discussion

Dans ce chapitre, nous avons présenté une revue de la littérature des différents algorithmes utilisés ou développés pour la reconstruction des images en TEP. Nous avons classifié ces algorithmes selon leur nature : algorithmes analytiques, itératifs déterministes, itératifs stochastiques. Nous avons aussi présenté très rapidement les algorithmes 4D utilisés pour construire les images des paramètres physiologiques directement à partir des données d'acquisition, en les comparant aux reconstructions 3D dynamiques qui permettent une reconstruction indirecte de ces

paramètres.

L'effort a été mis pour expliquer d'une manière simple tous ces algorithmes, discuter leurs forces, leurs limitations et des obstacles qui limitent l'utilisation de quelques-uns en clinique usuelle malgré qu'ils permettent une quantification plus précise.

Nous avons noté à travers cette recherche bibliographique que les algorithmes itératifs stochastiques sont les plus puissants et sont de plus en plus utilisés en clinique. En effet, ces derniers permettent une modélisation plus réaliste du processus d'acquisition, ce qui permet de bien contrôler le bruit des images construites et d'améliorer l'exactitude de la quantification. Mais, à cause de l'intensité de calcul de ces algorithmes, la reconstruction se fait surtout à partir des données d'acquisition en mode histogramme qui sont compressées et en utilisant des versions plus rapides de ces algorithmes tels que l'OSEM qui convergent vers des solutions biaisées. Par conséquent, on limite la précision de la quantification que théoriquement peuvent offrir les algorithmes stochastiques.

D'autre part, on reconstruit surtout des images statiques 3D afin d'estimer le paramètre SUV qui est un paramètre très sensible au processus d'acquisition et à l'algorithme de reconstruction utilisé. L'amélioration de l'exactitude de la quantification de SUV nécessite la reconstruction à partir des données non compressées et l'utilisation des algorithmes stochastiques non biaisés tels que MLEM pour les données stockées sous forme de sinogrammes et LM-EM pour des données en mode liste. Pour ceci, il est nécessaire d'utiliser des plates-formes de calcul parallèles pour réduire le temps de calcul de ces algorithmes à des niveaux acceptables cliniquement. Les cartes graphiques GPU offrent une solution peu onéreuse qui va permettre d'arriver à ce but.

Par ailleurs, le paramètre SUV ne permet pas d'extraire toute l'information clinique fonctionnelle et métabolique que potentiellement peut offrir la modalité fonctionnelle TEP. Pour que cette modalité devienne une modalité métabolique qui permet de faire de la biochimie *in vivo* en quantifiant d'une manière précise des paramètres physiologiques, le taux de métabolisme cellulaire et la densité des récepteurs, il faut extraire ces paramètres directement à partir des données en mode liste en utilisant les algorithmes stochastiques utilisant des fonctions de décomposition spatio-temporelles pour effectuer des reconstructions 4D. Les cartes graphiques GPU sont prometteuses pour réduire le temps de calcul de ces algorithmes stochastiques 4D afin de les introduire en utilisation clinique.

Deuxième partie

Matériels

CHAPITRE 5

CARTES GRAPHIQUES

Dans notre travail, les algorithmes de reconstruction développés ont été implémentés et accélérés sur le GPU Tesla C2050. Ce chapitre explique l'architecture physique des GPUs, leurs avantages par rapport aux CPUs dans le domaine de la reconstruction de l'imagerie médicale, les bases de leur programmation et les techniques d'optimisation des algorithmes implémentés sur ces dispositifs de calcul intensif.

5.0.1 Historique

Les premières cartes graphiques programmables ont été développées vers la fin des années 90 [150]. C'était un ensemble de pipelines graphiques qui permettent de décharger les ordinateurs des fonctions graphiques spécifiques telles que le rendu d'image 2D/3D, la fonction de rastérisation et fragmentation. Un pipeline contient une succession de circuits électroniques intégrés dédiés à réaliser chacune des fonctions graphiques. La programmation de ces dispositifs se faisait essentiellement par l'une des deux interfaces de programmation API (*Application Programming Interface*) graphiques Direct3D ou OpenGL.

Le développement rapide du marché des jeux vidéo a poussé les compagnies NVIDIA et ATI à développer début des années 2000 des GPUs selon une nouvelle architecture appelée *Unified Shading Architecture*. Dans ces nouveaux GPUs, les circuits dédiés pour effectuer les fonctions graphiques sur chaque pipeline ont été remplacés par un processeur générique qui effectue ces fonctions en ordre. Cette nouvelle architecture offre plus de fonctionnalités de calcul parallèle et plus de flexibilité dans la programmation de ces dispositifs. Des langages de programmation graphiques de haut niveau comme Cg de Nvidia (compatible OpenGL/DirectX) et HLSL de Microsoft (compatible API DirectX uniquement) ont été aussi développés pour rendre la programmation plus facile, plus rapide et transparente de la couche matérielle.

Les scientifiques ont vite saisi la possibilité qu'offrent ces nouvelles cartes GPUs pour accélérer les calculs par parallélisation. Cependant, la problématique résidait dans la difficulté de programmer ces dispositifs via des API graphiques qui sont surtout l'apanage des professionnels des multimédia. De ce fait, les chercheurs de l'université Stanford ont développé, en 2004, le

langage BrookGPU, qui est une extension de C permettant de créer une interface haut niveau avec les API graphiques OpenGL et DirectX. L'objectif était de faire d'un GPU un coprocesseur du CPU spécialisé dans les calculs parallèles. La principale limitation du langage Brook est le problème de compatibilité avec les nouvelles versions GPU. Mais, il a le mérite d'être le pionnier de l'architecture GPGPU (***General-Purpose Computing on Graphics Processing Units***) qui a fait du GPU un outil puissant pour accélérer les calculs dans plusieurs domaines, tels que le graphisme et l'imagerie médicale.

Cependant, l'engouement pour l'utilisation des GPUs pour effectuer des calculs scientifiques complexes a eu lieu à partir, principalement, de 2007 lorsque NVIDIA a introduit CUDA (***Compute Unified Device Architecture***) qui est une plateforme GPGPU. Cette dernière désigne une architecture matérielle et un environnement logiciel qui permet d'exécuter sur les processeurs GPU, d'une manière parallèle et efficiente, des programmes de calcul écrits dans des langages de haut niveau tels que C, C++, Fortran, DirectCompute et OpenCL. Depuis cette date, plusieurs librairies mathématiques ont été aussi développées pour faciliter la programmation CUDA [1]. Plusieurs travaux d'accélération des calculs sur GPU ont été aussi publiés dans différents domaines. Dans le domaine de la reconstruction des images en TEP, nous notons près d'une centaine de travaux publiés jusqu'à maintenant.

5.1 Architecture matérielle CUDA des GPUs Fermi

Le GPU est une carte électronique prériphérique (*Device* en anglais) insérable dans le bus PCIe de la carte mère d'un ordinateur CPU hôte (figure 5.1). Dans l'architecture CUDA, le GPU est composé essentiellement d'une puce (*chip*) contenant un certain nombre de multiprocesseurs SMPs (*Streaming Multiprocesseurs*), et une mémoire vive dynamique DRAM (*Dynamic Random Access Memory*) à l'extérieur de la puce qui contient les SMPs (*off-chip*). Chaque SMP contient un certain nombre de processeurs SPs (*Streaming Processor*).

Jusqu'à maintenant, NVIDIA a développé trois grandes classes de GPUs d'architecture CUDA ayant différentes capacités de calcul, telles que les architectures Tesla, Fermi et Kepler et qui supportent respectivement les capacités de calcul 1.x, 2.x et 3.x, où x est un nombre qui définit les sous-versions des GPUs reflétant des améliorations mineures par rapport à l'une des trois classes d'architectures principales. Par simplification du langage, nous allons, dans ce qui suit, désigner par exemple : "GPU ayant les capacités de calcul 1.3" par : " GPU 1.3". Le GPU Tesla C2050,

utilisé dans notre travail, a une architecture Fermi avec 14 SMPs et supporte les capacités de calcul 2.0. Comme le montre la figure 5.2, chaque SMP de l'architecture Fermi contient :

- 32 SPs dont chacun contient une unité ALU (*Arithmetic Logic Unit*) pour les opérations arithmétiques et logiques sur des entiers et une unité FPU (*Floating Point Unit*) pour les opérations en virgule flottante. Cette architecture permet à chaque SMP d'effectuer 32 opérations arithmétiques en simple précision (32 bits) par cycle d'horloge sur des entiers et des nombres en virgules flottantes, et 16 opérations sur les nombres en double précision (64 bits). Fermi supporte en simple et double précision le nouveau standard IEEE 754-2008 qui permet de fusionner les opérations d'addition et multiplication dans une seule opération (FMA : *Fused Multiply-Add*). Un FMA calcule en une seule instruction $D = A * B + C$ sans perte de précision.
- 4 unités de fonctions spéciales SFUs (*Special Function Unit*) pour exécuter les fonctions transcendantes comme le cosinus, le sinus, la racine carrée et leur inverse. Chaque SFU exécute une opération par fil d'exécution (*thread*) par cycle d'horloge.
- 32768 registres de 32 bits (32 ko), un cache pour les instructions et deux unités de planification et de distribution des fils d'exécution (*two multithreaded warp schedulers and instruction dispatch units*). Chaque SMP exécute en concurrence deux groupes de 32 fils d'exécution chacun qu'on appelle *warps*. Chaque unité *warp scheduler* sélectionne un *warp* et envoie une instruction d'exécution à 16 SPs. Les 32 fils d'exécution de chaque *warp* sont exécutés en 2 et 4 cycles respectivement pour les opérations simples et doubles précisions.
- 16 unités de lecture et d'enregistrement (*load/store units*) permettant de calculer par cycle d'horloge 16 positions d'adresses mémoires.
- 64 ko de mémoire cache de premier niveau par SMP et qui est configurable pour être partagée entre la mémoire cache L1 et la mémoire partagée (*shared memory*). L'utilisateur peut attribuer 16 ko à L1 et 48 ko à la mémoire partagée ou 48 ko à L1 et 16 ko à la mémoire partagée. L1 permet d'offrir à chaque SMP un cache pour l'accès plus rapidement à la mémoire DRAM et aux registres temporaires résidant dans la mémoire DRAM. Lorsqu'elle est utilisée correctement, la mémoire partagée permet d'accélérer l'accès à la mémoire DRAM.
- Une mémoire L2 (*uniform cache*) de deuxième niveau de capacité 768 ko qui sera utilisée comme cache de la mémoire DRAM pour tous les SMPs. L2 offre au SMP un accès rapide

à la mémoire DRAM pour les requêtes de lecture et d'enregistrement, et pour les demandes de service de texture et des opérations atomiques. L2 permet de diminuer énormément la latence d'accès à la mémoire DRAM, surtout pour les applications où l'accès à la mémoire DRAM se fait d'une manière aléatoire, c'est à dire les applications où les adresses ne sont pas connues auparavant pour optimiser l'accès par coalescence, ce qui est le cas de la reconstruction des images TEP à partir des données en mode liste.

Les fonctions atomiques permettent à un seul fil d'exécution de lire une donnée à partir d'une position mémoire, de la modifier et puis d'écrire le résultat dans la même case mémoire. Les autres fils d'exécution ne peuvent accéder à cette case mémoire que lorsque la boucle (lire, modifier et écrire) est complétée par le premier fil d'exécution. Parmi ces fonctions, on cite la fonction *atomicAdd()* qui permet de modifier le contenu, en virgule flottante, d'une case mémoire par une addition d'une autre valeur. Cette dernière est très importante dans la reconstruction des images en TEP, car elle permet de réaliser une rétroprojection des données d'acquisition dans l'image sans perte. Grâce à l'implantation au niveau matériel de plus d'unités de calcul atomique et l'utilisation de la mémoire L2, l'architecture Fermi permet d'accélérer les fonctions atomiques de plus de 20 fois par rapport aux GPUs de la génération précédente.

Figure 5.1 – Image de la carte GPU Tesla C2050.

La communication entre la puce GPU et sa mémoire DRAM se fait via 6 contrôleurs haute vitesse GDDR5 (*Graphics Double Data Rate, version 5*) de 64 bits chacun (bus de 512 bits) permettant une grande bande passante théorique de 140 Go/s. Par contre la bande passante de la communication entre la mémoire DRAM de GPU et la mémoire RAM CPU de l'hôte à travers

le bus PCI express x16 Gen2 est de 8 Go/s. Par ailleurs, Fermi a un bus d'adresses de 40 bits qui permet d'adresser 1 To d'espace mémoire continuellement dans l'espace des adresses de GPU, de CPU et de celui des autres périphéries PCIe.

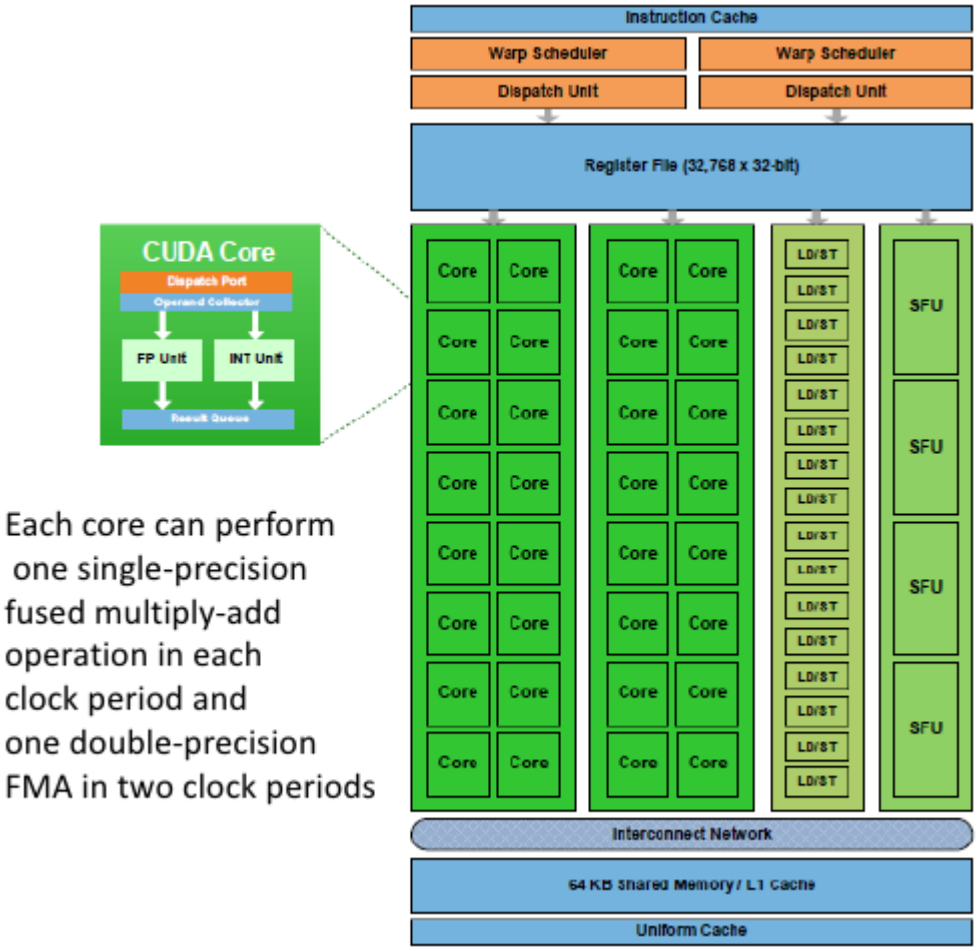


Figure 5.2 – Architecture Fermi des multiprocesseurs SMP. Source : [150].

5.2 Performances de calcul des GPUs

Les GPUs ont une puissance de calcul et une bande passante d'accès à leur mémoire DRAM très élevée en comparaison aux CPUs comme le montrent les figures 5.3 et 5.4 : La carte GPU Tesla C2050 qui a 448 SPs a une puissance de calcul théorique de 1.03 Tflops/s en virgule flottante simple précision (tableau 5.I), alors que le CPU Intel Westmere développé la même année que le Tesla C2050 a une puissance théorique de calcul de 160 Gflops/s. Le CPU Westmere

Tableau 5.I – Les principales caractéristiques de la carte GPU Tesla C2050.

Nombre de SMP :	14	Performances en virgule flottante simple précision :	1,03 Tflops/s	Mémoire DRAM :	3 Go GDDR5
Nombre de SP :	448	Performances en virgule flottante double précision :	160 Gflops/s	Bande passante GPU-DRAM :	144 Go/s
Nombre maximal de fils d'exécution par SMP :	1536	Nombre de registre par multiprocesseur :	32 k	Capacités de calcul :	2.0
Interface GPU-CPU :	PCIe x16 Gen2	Bande Passante GPU-CPU :	8 Go/s	Consommation maximale :	238 W

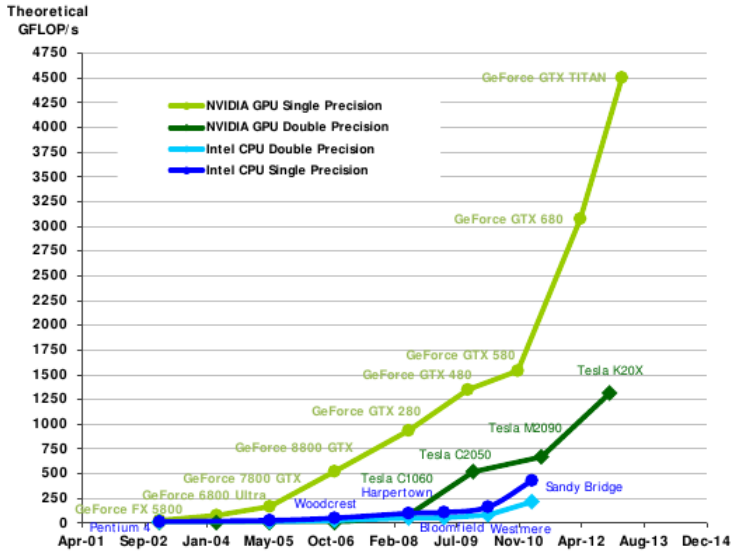


Figure 5.3 – Puissance de calcul des GPU et des CPU en Gflops/s pour les opérations sur les nombres réels en virgule flottante. Source : Nvidia [151].

contient 6 cœurs avec une fréquence d'horloge de 3 Mhz. Les GPUs sont donc la plateforme de calcul qui offre la puissance de calcul la moins chère et aussi la plus économe en termes de consommation électrique. Schellmann et al. [198] ont montré qu'une grappe de 4 cartes GPU NVIDIA GeForce 8800 GTX d'un coût estimé à 6000 $ offre presque la même vitesse de reconstruction en TEP à partir des données en mode liste qu'une grappe de 200 nœuds de CPU Dual INTEL Xeon 3.2 GHz 64 bit qui coûte 2 250 000 $.

Cette différence dans la puissance de calcul s'explique par le parcours historique de développement des GPUs et des CPUs. Les GPUs, qui sont destinés au départ aux traitements spécialisés de graphisme et des jeux vidéo, se sont basés sur une architecture hautement parallèle (plus de 100 pipelines) qui applique d'une manière synchrone la même opération sur des données multi-

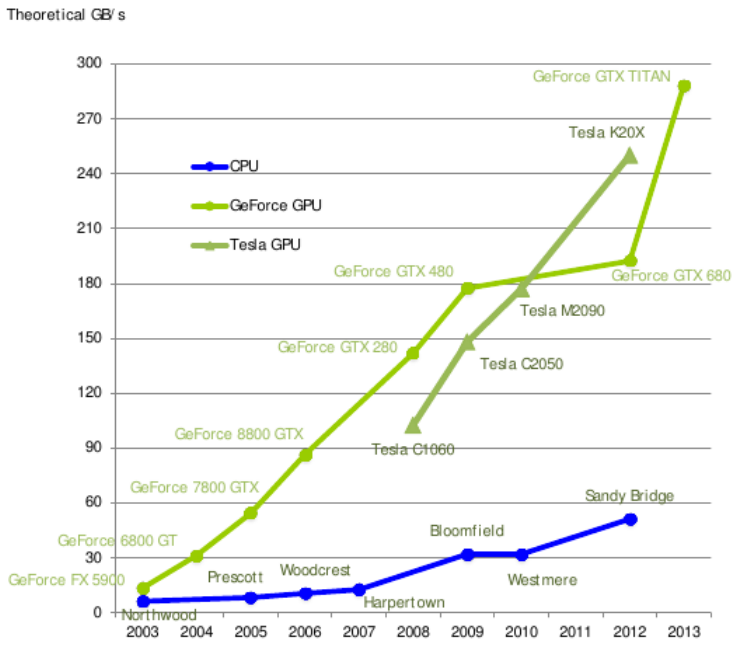


Figure 5.4 – Bande passante des GPU et des CPU avec leur mémoire DRAM en Goctets/s. Source : Nvidia [151].

ples de la même nature (programmation SIMD : *Single Instruction Multiple Data*). Ces données sont stockées dans une mémoire avec une grande bande passante, car l'accès à ces données se fait d'une façon cohérente. Donc, le nombre de transistors destiné aux calculs est largement supérieur au nombre de transistors dédiés aux unités qui contrôlent le flux et celles qui réalisent les caches pour les données (figure 5.5). À l'inverse, le CPU est dédié au début pour effectuer

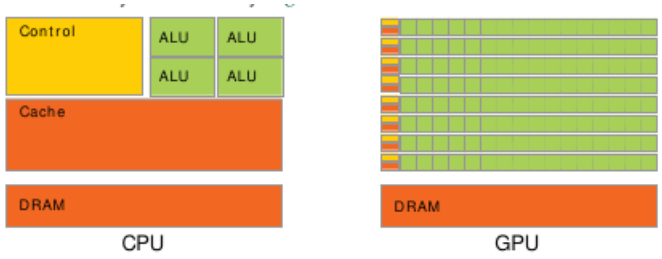


Figure 5.5 – Schéma montrant la densité des transistors destinés aux calculs et ceux dédiés aux contrôles de flux dans les GPU et les CPU. Source : Nvidia [151].

différentes instructions sur des données diverses (entiers, flottants) et dont l'accès dans la mémoire est généralement aléatoire. Donc le nombre de transistors dédiés aux unités arithmétiques et logiques est moins important que le nombre de transistors utilisés dans les registres, les mé-

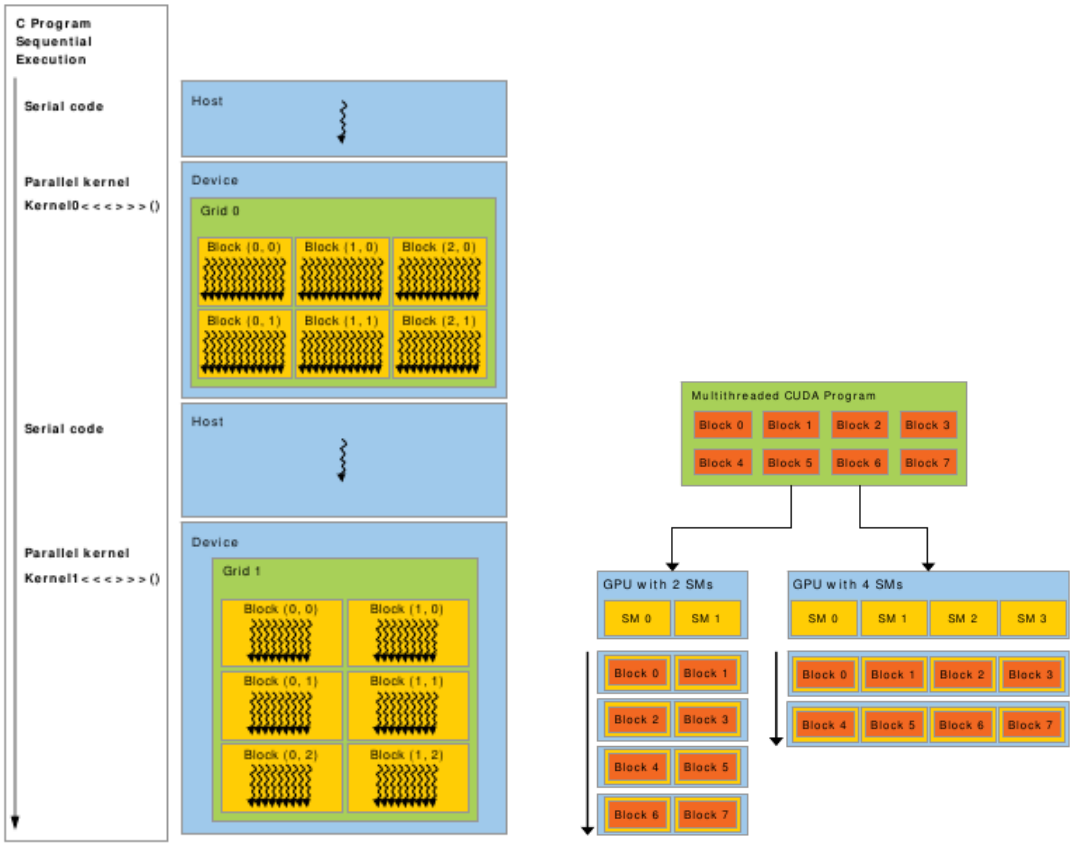
moires caches et les interfaces de contrôles. Pour augmenter les performances des CPUs, les concepteurs ont surtout travaillé cette dernière décennie à augmenter la fréquence de l'horloge qui vient d'atteindre les limites physiques de la dissipation de la chaleur. Ils ont aussi augmenté le nombre d'unités de traitement (CPU multi-cœurs) pour exécuter en parallèle plusieurs instructions. Mais le gain en puissance de calcul en parallélisant un flux séquentiel d'instructions est limité par un faible taux d'occupation des unités d'exécution.

Bien que les performances des GPU continuent d'augmenter rapidement en comparaison de celles des CPUs (figure 5.3), dont l'évolution semble atteindre la limite de la loi de Moore, les deux technologies ont tendance à converger vers les unités de calcul APU (*Accelerated Processing Unit*) qui intègrent sur la même puce un CPU multi-cœurs et un GPU en partageant le même espace mémoire.

5.3 Modèle CUDA de programmation

Une application CUDA est une suite d'instructions qui s'exécutent séquentiellement sur CPU et qui lance l'exécution des fonctions appelées *kernel* sur GPU. L'exécution de chaque *kernel* se fait en parallèle sur plusieurs fils d'exécution selon le paradigme SIMT (*Single Instruction Multiple Thread*). Comme le montre la figure 5.6 a, les fils d'exécution sont groupés dans un ensemble de blocs formant une grille (*grid*). Chaque *kernel* s'exécute sur une grille d'un ensemble de blocs. L'exécution des blocs de fils d'exécution se fait sans ordre précis dans les multiprocesseurs d'une manière concurrentielle et indépendamment les uns des autres (figure 5.6). L'exécution des blocs ne peut pas donc être synchronisée et les fils d'exécution de différents blocs ne peuvent pas partager les données. L'indépendance des blocs permet une indépendance (*scalability*) par rapport au nombre des SMPs. De ce fait, d'une part, l'utilisateur se concentre seulement sur la parallélisation au niveau logique d'un algorithme et non pas sur la gestion de la distribution des blocs de fils d'exécution sur les multiprocesseurs, et, d'autre part, les programmes seront compatibles avec les nouveaux GPUs qui arrivent sur le marché.

Les fils d'exécution d'un même bloc s'exécutent sur un SMP sous forme de *warp* qui est un groupe de 32 fils d'exécution d'indices consécutifs, et qui s'exécutent simultanément sur le SMP. L'unité de contrôle des multiprocesseurs crée, gère et planifie le séquencement (*scheduling*) des fils d'exécution et leur exécution sous forme de *warp*. Les fils d'exécution de chaque bloc peuvent être synchronisés et peuvent partager les mêmes données à travers la mémoire partagée.



a b

Figure 5.6 – Principe du modèle de programmation *Single Instruction Multiple Thread* sur GPU : a) organisation des fils d'exécution et b) exécution concurrente des blocs de fils d'exécution sur les SMPs.

Le nombre de fils d'exécution par blocs est limité, puisque que les fils d'exécution du même bloc partagent les ressources du même SMP, à savoir les registres et la mémoire partagée (section 5.4). Il est, par exemple, de 1024 fils d'exécution pour les GPUs ayant des capacités de calcul 2.x et 512 fils d'exécution pour les versions antérieures 1.x.

5.4 Organisation de la mémoire en CUDA

Durant l'exécution d'un *kernel* sur un périphérique GPU, les fils d'exécution ont accès à des données dans différents types de mémoires comme le montre les figures 5.7 a et b. Chaque espace mémoire se caractérise par sa localisation physique (sur la puce GPU ou sur la barrette DRAM

de GPU), sa visibilité, son type d'accessibilité (lecture, écriture ou les deux) et sa durée de vie. Ces espaces mémoires sont :

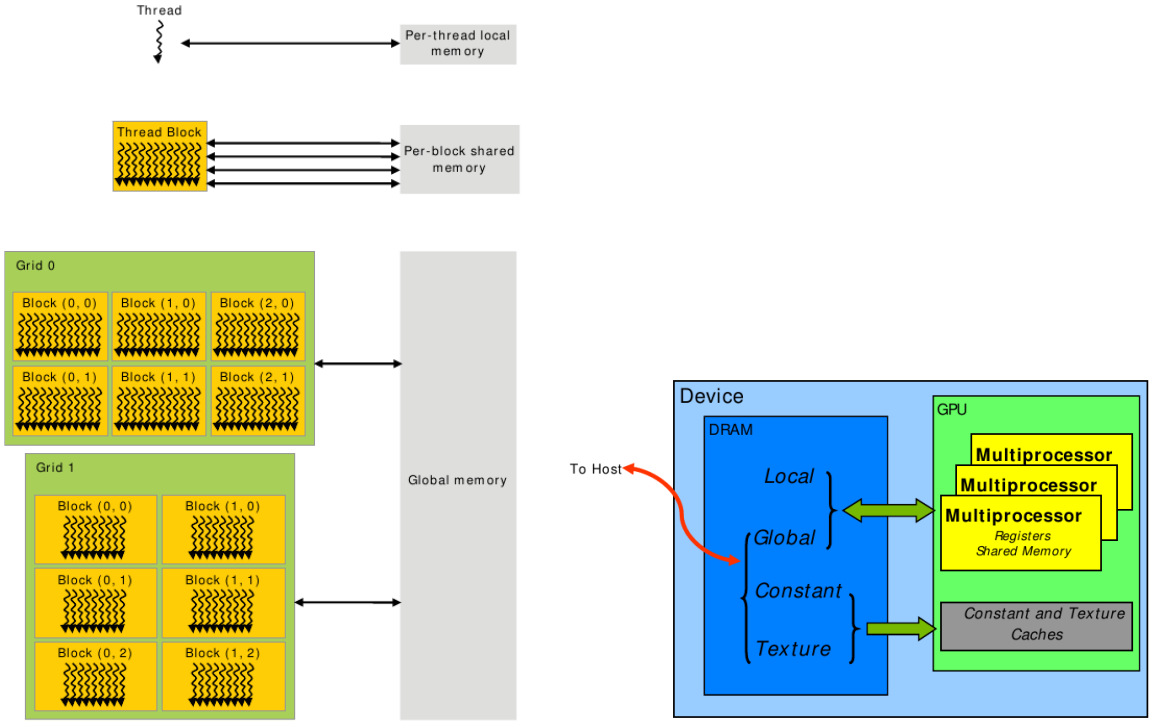


Figure 5.7 – Architecture CUDA : a) modèle hiérarchique de la mémoire et b) localisation des différentes types de mémoires.

1. **Registres et mémoire locale** : les registres sont localisés dans la puce GPU et chaque fil d'exécution a accès à ses propres registres privés. Puisque le nombre maximal de registres par SMP est limité et que les registres sont alloués à tous les fils d'exécution actifs, des registres, au besoin, seront localisés dans la mémoire locale qui réside dans DRAM. Les registres et la mémoire locale sont des positions privées pour chaque fil d'exécution et qui seront libérés après son exécution. Par ailleurs, l'accès à la mémoire locale est très long en comparaison aux registres, à la mémoire partagée, et à la mémoire globale. Pour les GPUs 2.x comme la Tesla C2050, la mémoire locale est mise en cache dans L1 et L2, ce qui permet d'accélérer son accès en comparaison aux GPUs 1.x.
2. **Mémoire partagée** : comme nous avons déjà expliqué à la section 5.1, c'est une mémoire intégrée dans la puce GPU, et qui est organisée pour que chaque SMP ait son propre espace

mémoire partagée. L'espace alloué à chaque SMP est accessible seulement aux fils d'exécution du bloc actif pour les requêtes de lecture et d'écriture, et elle est libérée lorsque tous les fils d'exécution du bloc ont été exécutés. Parce qu'elle est localisée sur la puce GPU, la mémoire partagée est aussi rapide que les registres lorsque les conflits d'accès sont évités. En effet, elle est divisée en 32 banques qui sont organisées pour que des mots de 32 bits successifs soient assignés à des banques successives. De ce fait, les fils d'exécution du même *warp* peuvent accéder simultanément aux 32 mots s'il n y pas des conflits de banques. Pour les GPUs 2.x, le conflit aura lieu si deux ou plusieurs fils d'exécution accèdent simultanément à la même banque. Dans cette situation le transfert des données sera séquentiel au niveau de la banque sujette à un ou plusieurs conflits.

3. **Mémoire globale** : elle est configurée sur DRAM de GPU et elle est accessible à tous les fils d'exécution des différents *kernels* exécutés sur GPU. Elle est aussi accessible au CPU pour y exécuter les requêtes de lecture et d'écriture (figure 5.7). La mémoire globale est la mémoire la plus considérable, la plus sollicitée durant l'exécution des *kernels* et elle reste persistante entre les différents *kernels* envoyés à l'exécution sur GPU tout le long de l'application. La bande passante théorique de l'accès du CPU à la mémoire globale est limitée (8 Go/s pour l'architecture Fermi). Par ailleurs, la bande passante théorique de transfert des données entre le GPU et la mémoire globale est très grande (144 Go/s), mais cette bande passante reste très inférieure, par un facteur de l'ordre de 100, par rapport à celle des registres et de la mémoire partagée, et elle décroît rapidement si les conditions de coalescence ne sont pas respectées. Le fait de ne pas maximiser l'accès à la mémoire globale a un grand impact sur le temps d'exécution des *kernels*. Pour les GPUs Fermi, la mémoire globale bénéficie de la présence des mémoires tampons L1 et L2, ce qui permet de réduire le temps d'accès pour les données en comparaison aux architectures précédentes.
4. **Mémoire texture :** elle est aussi configurable dans l'espace mémoire physique DRAM. Elle est accessible à l'hôte CPU comme mémoire globale pour y écrire des données et au GPU comme texture pour y lire des données (*fetch data*). Tous les fils d'exécution des grilles sont accessibles à la mémoire texture qui reste persistante durant toute l'application. La mémoire texture utilise une mémoire cache en lecture. Elle permet donc de réduire et de maintenir constante la latence d'accès à DRAM s'il y a localité des données, c'est-à-dire les fils d'exécution du même bloc utilisent des données localisées dans un voisinage spatial sur DRAM. De plus, la lecture des textures (*fetch*) se fait via une unité matérielle

d'interpolation linéaire dédiée.

5. **Mémoire constante :** elle est localisée aussi dans l'espace mémoire DRAM, et elle est accessible par le CPU pour écriture et par tous les fils d'exécution des *kernel* pour la lecture seulement. Elle est très limitée puisque sa taille est de 64 Ko par GPU et elle est benéficie d'une mémoire cache de même taille pour permettre un accès aussi rapide que les registres.

5.5 Interface CUDA de programmation

Deux plates-formes de programmation se sont imposées pour programmer les GPUs : 1) OpenCL (*Open Computing Language*) développé par consortium de compagnies entres autre Apple pour programmer des systèmes parallèles hétérogènes comprenant à la fois des CPUs multi-cœurs et des GPUs, et 2) CUDA C de NVIDIA que nous avons utilisé pour effectuer ce travail, et que nous présentons brièvement ci-dessous.

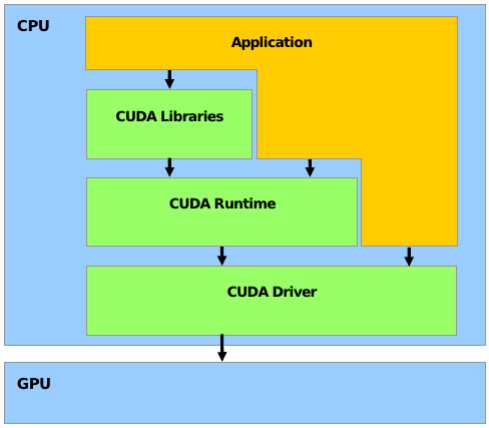


Figure 5.8 – Structure de la plateforme NVIDEA CUDA C. Source : Nvidia [151]

La plate-forme CUDA C est constituée d'une interface de programmation API CUDA C, d'un pilote (*Driver*) GPU, des librairies d'application et d'un compilateur nvcc (figure 5.8). L'API CUDA C est une extension du langage ANSI C permettant d'écrire les *kernels* qui seront exécutés sur GPU. Elle est constituée aussi de la librairie *Runtime* qui est un ensemble de fonctions C qui s'exécute sur l'hôte CPU. Ces fonctions permettent d'allouer et de libérer la mémoire sur le GPU, de transférer les données entre l'espace mémoire CPU et GPU, de gérer les textures, l'exécution des *kernels* et les erreurs sur GPU, d'initialiser le GPU et de gérer les systèmes avec plusieurs

GPUs, etc. (le document [152] détaille toutes les fonctions *Runtime*).

Une application CPU-GPU est un mélange d'instructions qui seront exécutées sur CPU, des fonctions écrites par l'utilisateur et celles de la librairie *Runtime* qui seront exécutées aussi sur l'hôte CPU, et aussi des *kernels* développés par l'utilisateur pour être exécutés sur le périphérique GPU. Lors de la compilation, nvcc sépare les codes des deux plates-formes d'exécution et ensuite :

- compile les *kernels* GPU en code assembleur PTX (*Parallel Thread eXecution*) et/ou au binaire (objet Cubin) ;
- modifie le code qui sera exécuté par l'hôte en remplaçant des directives pour l'exécution des *kernels* sur GPU par des fonctions *Runtime* qui prendront en charge la lecture et le lancement de ces *kernels* compilés. Après modification, ce programme devient donc un programme ANSI C qui sera soit compilé directement par un compilateur du langage C invoqué par nvcc, soit il sera compilé indépendamment par l'utilisateur.

Pour exécuter les *kernels* sur GPU dans le cas où ils sont compilés seulement en assembleur PTX, le *Runtime* charge ces *kernels* et les compile par le pilote du GPU avant de les lancer pour l'exécution. On appelle cette approche *Just-in-time compilation*. Ensuite, le fichier des *kernels* compilés en binaire sera gardé en cache pour les futures exécutions de l'application afin d'éviter de répéter la compilation à chaque exécution. La compilation des *kernels* sera réeffectuée seulement s'il y a l'installation d'un nouveau pilote GPU. *Just-in-time compilation* a l'inconvénient d'alourdir un peu le lancement des *kernels*, mais elle permet de profiter des améliorations des nouveaux pilotes GPU et de pouvoir exécuter ces *kernels* sur les nouveaux GPUs qui vont arriver sur le marché.

Par ailleurs, les *kernels* compilés en PTX ou en binaire peuvent être aussi chargés et lancés sur GPU par une application hôte écrite par l'interface de programmation de bas niveau *Driver API*. Cette API offre les mêmes fonctions que l'API *Runtime*, mais en permettant un meilleur contrôle sur l'exécution des *kernels* sur GPU que cette dernière. Son inconvénient est qu'elle est difficile à programmer et à débugger. Une application CPU-GPU doit soit utiliser CUDA *Runtime* API, soit CUDA *Driver* API. Les deux environnements ne peuvent être utilisés en même temps.

NVIDIA offre aussi un ensemble d'utilitaires pour faciliter la programmation, tel que NVIDIA *Visual Profiler* qui permet d'analyser le temps d'exécution des différents éléments du programme pour faciliter l'optimisation, *CUDA-GDB* pour débugger les applications CPU-GPU, et *CUDA-MEMCHECK* qui permet d'analyser les erreurs mémoires. Des librairies mathématiques opti-

misées pour simplifier le développement des applications CPU-GPU sont aussi téléchargeables avec la plate-forme CUDA C. Parmi elles, il y a :*CUDA SDK, CUFFT, CUBLAS, Thrust, PhysX* et *OpticX*.

5.6 Optimisation de la programmation en CUDA

Les GPUs utilisent une parallélisation massive qui permet d'offrir une grande puissance de calcul. Mais tirer profit de cette puissance nécessite un grand travail d'optimisation de l'implémentation des *kernels*. L'optimisation doit se baser sur une bonne connaissance de l'architecture CUDA du GPU utilisé, sinon les performances obtenues en pratique seront très décevantes. On doit surtout optimiser la collaboration CPU GPU, maximiser l'occupation du GPU, optimiser l'accès et l'utilisation des différents types de mémoires, minimiser l'utilisation des fonctions coûteuses en temps d'exécution, minimiser l'utilisation des instructions conditionnelles, utiliser des librairies optimisées.

5.6.1 Collaboration CPU GPU

Comme il a été déjà expliqué le GPU agit comme un coprocesseur pour le CPU dont la mission est d'accélérer l'exécution des *kernels*. Mais pour atteindre les performances possibles par le matériel, un effort d'analyse de l'application à mettre en oeuvre est nécessaire afin de déterminer les fonctions qui seront exécutées sur GPU sous forme de *kernels* et celles qui seront exécutées sur CPU. Les *kernels* GPU doivent porter surtout sur les données indépendantes et sur lesquelles le même traitement est appliqué. Les meilleures performances sur GPU sont obtenues, généralement, s'il y a suffisamment de fils d'exécution qui s'exécutent en concurrence sur tous les processeurs SPs du GPU et si la collaboration et l'échange des résultats intermédiaires entre fils d'exécution est faible. Les fonctions qui ne permettent pas une parallélisation intense ou qui utilisent beaucoup les instructions de branchements pourraient s'exécuter avec une meilleure performance sur les CPU multi-coeurs.

D'autre part, l'exécution des *kernels* est asynchrone, c'est-à-dire que le CPU prend le contrôle rapidement juste après le lancement de kernels, et que le transfert de données entre l'espace mémoire de CPU et celui de GPU peut se faire aussi d'une façon asynchrone. Donc, il serait plus rentable que le code implanté permette d'exécuter simultanément des fonctions sur CPU et des *kernels* sur GPU.

5.6.2 Occupation des multiprocesseurs

L'occupation est définie comme le rapport entre le nombre de *wraps* actifs par SMP et le nombre de *warps* maximal possible par SMP. Elle dépend énormément du nombre de registres que chaque fil d'exécution utilise et du nombre de fils d'exécution par bloc fixé par l'utilisateur. En effet, les registres et la mémoire partagée sont alloués à tous les fils d'exécution de tous les blocs actifs pour permettre un ordonnancement rapide de tous les *warps* résidant dans le SP. Pour respecter les contraintes des sources mémoires disponibles, le nombre de blocs concurrents est choisi dynamiquement par l'unité *multithreaded warp schedulers* et l'occupation diminuera dans le cas des grands blocs.

Pour clarifier ce dernier paragraphe, considérons que l'utilisateur a choisi des blocs de taille 800 fils d'exécution pour exécuter un *kernel* sur GPU Tesla C2050. Puisque, pour cette architecture, chaque fil d'exécution utilise 21 registres de 32 bits et que le maximum de registres par SMP est 32768 (tableau 5.I), alors le nombre de blocs qui seront actifs dans un SMP est 1. En effet, 2 blocs nécessitent 33600 registres qui est un nombre supérieur à la valeur maximale de registres par SMP. Le nombre de fils d'exécution actifs par SMP sera donc 800 fils d'exécution et, par conséquent, l'occupation est 800/1536= 52%. 1536 est le nombre maximal des fils d'exécution actifs par SMP pour les GPU de génération 2.x.

Par ailleurs, une grande valeur d'occupation ne signifie pas un très grand rendement de GPU, il y a un point au-dessus duquel l'augmentation de l'occupation des SMPs n'améliore pas ce rendement. Mais une faible occupation détériore les performances de calcul. Il faut s'assurer d'avoir une occupation supérieure à 66% [151]. Pour ceci, NVIDIA préconise : i) que le nombre de fils d'exécution par bloc soit un multiple de la taille du *warp* (32 pour les GPU 2.x), afin d'éviter que des *warps* contiennent des fils d'exécution inactifs et de faciliter la coalescence d'accès à la mémoire partagée et la mémoire globale, ii) de s'assurer d'avoir au moins 3 blocs actifs par SMP et un minimum de 64 fils d'exécution par blocs, et iii) de choisir comme départ pour des tests d'optimisation un nombre de fils d'exécution par bloc qui est entre 128 et 256.

Afin de faciliter l'optimisation, NVIDIA fournit un outil Excel qui calcule l'occupation des SMPs en fonction des différents paramètres du *kernel* : génération du GPU, nombre de registres utilisés par fil d'exécution, nombre de fils d'exécution par bloc et mémoire partagée par bloc.

5.6.3 Optimisation d'accès aux mémoires

Le plus grand défi dans le développement des applications qui exécutent des *kernels* sur les périphériques GPU est l'optimisation de l'accès aux différents espaces mémoires. L'efficience de l'exécution des *kernels* sur GPU dépend énormément de la latence de transfert des données entre les différents types de mémoires qui sont localisées dans l'espace physique DRAM et la puce GPU. Pour augmenter le débit du transfert des données aux SMPs, il faut :

- Minimiser le transfert de données entre l'espace mémoire CPU et l'espace mémoire GPU en transférant le maximum de code de CPU vers le GPU, même s'il se traduit par une faible parallélisation sur GPU. Tous les résultats intermédiaires de l'exécution d'un *kernel* sur GPU, doivent être générés, utilisés et détruits dans l'espace mémoire GPU. Il est aussi plus rentable de faire le transfert de toutes les données de la mémoire CPU vers la DRAM du GPU en une seule opération que de le diviser en plusieurs petits transferts. En outre, pour les applications ayant beaucoup de données à transférer entre CPU et le GPU comme pour la reconstruction TEP à partir du mode liste, il est très gagnant de chevaucher le transfert de données et l'exécution du *kernel* en les divisant en plusieurs parties associées à différentes flots (*streams*) et en utilisant le transfert asynchrone.
- S'assurer que l'accès à la mémoire globale du GPU se fait en respectant les contraintes de coalescence (voir le document [151] pour les détails sur les conditions de coalescence). Si les conditions de coalescence ne sont pas respectées, la lecture des données pour un *wrap* se fait en plusieurs accès au lieu d'un seul de 128 octets, ce qui dégrade énormément les performances d'exécution de GPU. L'utilisation des structures de tableaux (SoA : *Structure of Arrays*) au lieu des tableaux de structures (AoS :*Arrays of Structure*s) augmente le flux d'accès à la mémoire globale en permettant de respecter facilement les contraintes d'alignement.
- Utiliser d'une façon judicieuse la mémoire partagée. En effet, cette mémoire offre à la mémoire globale un cache contrôlable et aussi rapide que les registres lorsque les conflits des banques sont évités. Mais son utilisation a un impact sur l'occupation des SMPs. L'utilisation de la mémoire partagée est surtout rentable lorsque l'accès à la mémoire globale est redondant comme dans le cas de la multiplication de deux matrices ; l'accès aux lignes et aux colonnes de deux matrices à multiplier est répété.
- Utiliser les textures lorsque : i) il n'est pas possible de remplir les conditions de la coales-

cence pour la mémoire globale, car, pour les GPU 2.x et 3.x, la mémoire globale bénéficie des mémoires tampons et, par conséquent, présente une meilleure bande passante que la texture si les conditions d'alignement (coalescence) ont été respectées, ii) les adresses de la texture ne sont pas déterminées avant l'exécution du *kernel* ou bien elles sont calculées à l'extérieur du *kernel* par des unités dédiées, iii) la valeur de la texture retournée est un entier qui sera converti en un nombre réel simple précision de l'intervalle [0,1] ou [-1,1], et iv) les coordonnées de la texture sont réelles et donc la valeur retournée sera interpolée à partir des données de la matrice des données puisque les GPUs utilisent pour les textures une interpolation bilinéaire implémentée au niveau matériel.

5.6.4 Maximisation du flux d'instructions

Les GPUs ont adopté le standard IEEE 754 dans la représentation des nombres réels. Les GPUs 1.2 et les versions antérieures supportent seulement les réels en virgule flottante codés en simple précision sur 32 bits, alors que les nouvelles versions ont intégré aussi les réels en double précision sur 64 bits. Ces derniers sont beaucoup plus coûteux en temps de calcul que les calculs en simple précision [151]. Par exemple, pour le Tesla C2050 , les opérations standards d'addition, de multiplication, et de multiplication et addition combinées nécessitent chacune 1 cycle d'horloge en simple précision et 2 cycles en double précision.

Par ailleurs, en plus des librairies mathématiques simple et double précision, CUDA utilise une librairie intrinsèque des fonctions qui ne sont exécutables que sur GPU. Les opérations de cette librairie sont beaucoup plus rapides en exécution, mais moins précises que les fonctions standards. Pour augmenter le flux d'instructions, il est recommandé d'utiliser les réels en simple précision et les fonctions intrinsèques. Le compilateur nvcc permet d'appliquer une directive pour lui indiquer de remplacer les fonctions standards par les fonctions intrinsèques.

Il est aussi suggéré d'éviter d'appliquer des opérations arithmétiques sur des variables de type *char* ou *short*. Le compilateur introduit des instructions de conversions de ces variables en type *int* et augmente le nombre de cycles durant l'exécution. De la même manière, les constantes réelles en doubles précision augmentent le nombre de cycles lorsqu'elles sont utilisées dans les opérations en simples précision. L'exécution des fonctions atomiques est plus longue car l'accès à la position mémoire est sérialisé en cas de conflits entre les fils d'exécution, il faut donc éviter de les utiliser si l'impact sur la précision du calcul est minime.

Outre cela, l'utilisation des instructions de branchement *if, switch, do, for* et *while* pourrait

énormément dégrader le flux d'instructions. En effet, dans le cas où des fils d'exécution du même *wrap* divergent en suivant différentes branches, l'exécution se fait séquentiellement. Pour éviter ce problème, il est conseillé d'appliquer des stratégies qui regroupent les fils d'exécution qui suivent la même branche dans le même *warp*.

Il faut aussi éviter de faire appel intensivement à l'instruction *__syncthreads()* qui synchronise l'exécution des fils d'exécution du même bloc en jouant le rôle d'une barrière qui empêche de continuer l'exécution avant que tous les fils d'exécution du bloc aient fini l'instruction en cours. En effet, son utilisation pourrait faire chuter l'occupation des SMPs. Elle fait aussi augmenter inutilement le nombre de cycles d'exécution, car son flux d'exécution est de 8, 16 et 128 instructions par cycle d'horloge par périphérie pour, respectivement, les GPUs de capacités de calcul 1.x, 2.x et 3.x. L'utilisation des librairies optimisées citées auparavant à la section 5.5 permet aussi d'améliorer les performances d'exécution des fils d'exécution sur les GPUs.

5.7 Utilisation des GPUs dans la reconstruction de l'imagerie médicale

Avec les développements technologiques, les applications cliniques en imagerie médicale sont de plus en plus nombreuses et variées et les exigences en termes de qualité et du temps de production des images tomographiques sont de plus en plus sévères. Les professionnels de la santé veulent des modalités d'imagerie qui permettent de produire des images diagnostiques de plus en plus précises. Ils veulent aussi que ces images soient produites plus rapidement afin d'augmenter le débit des patients, ou même en temps réel afin d'effectuer de l'imagerie interventionnelle qui guide les gestes des chirurgiens en salle d'opération et d'améliorer la conformité de la radiothérapie en utilisant la technique de guidage par imagerie. Les modalités utilisant la radiation ionisante telles que la TDM, la TEP et la TEMP doivent aussi réduire la dose donnée aux patients pour se conformer aux normes de radioprotection dont les seuils sont en plus plus exigeants tout en produisant des images avec un meilleur RSB.

Pour répondre à ces exigences, les manufacturiers développent des équipements performants qui produisent des données d'acquisition de plus en plus volumineuses par l'amélioration de l'échantillonnage spatial en utilisant des détecteurs de petites tailles. Ils utilisent aussi des algorithmes de reconstruction complexes, comme les algorithmes itératifs stochastiques, pour améliorer le RSB des images et réduire la dose aux patients. Mais ceci nuit à l'autre exigence qui est d'effectuer des reconstructions ultra-rapides malgré l'utilisation d'ordinateurs de plus en plus puis-

sants.

Les algorithmes de reconstruction sont généralement de type flux de données sur lesquelles les mêmes opérations de projection et de rétroprojection sont appliquées simultanément en itérations. Ces algorithmes sont adaptés à une parallélisation intense pour être exécutés avec des bonnes performances sur des plates-formes de calcul parallèle, comme les ASICs (*Application Specific integrated Circuits*)[228], les FPGAs (*Field Programmable Gate Arrays*) [119] et les GPUs. Ces derniers ont suscité beaucoup d'intérêt dans la communauté scientifique pour accélérer et améliorer la reconstruction en imagerie médicale, car ils offrent un rapport performance/coût nettement supérieure aux autres dispositifs . Ainsi, plusieurs travaux ont été effectués pour accélérer et utiliser des algorithmes de reconstruction itératifs en TEP [13, 14, 24, 45, 57, 102, 145, 170–172, 198, 251], en TEMP [9, 137, 138, 214, 225, 235], TDM [22, 109, 148, 200, 220, 246, 250], en tomodensitométrie à géométrie conique (CBCT : *Cone Beam Computed Tomograph* [93, 160, 244, 248, 255], en imagerie par résonance magnétique (IRM) [79, 83] et en échographie [11, 103]. Les deux articles d'Eklund et al. [54] et de Pratx et al. [174] passent en revue l'utilisation actuelle et future des GPUs en imagerie médicale.

Malgré les nombreux travaux effectués par les chercheurs sur la reconstruction d'images médicales sur GPU, l'utilisation des GPUs pour améliorer la qualité de l'image en utilisant des algorithmes plus performants de reconstruction n'est pas introduite, pour le moment, en clinique par les manufacturiers. Cependant, des annonces ont été faites dans les derniers congrès de la *Radiological Society of North America* (RSNA) par quelques fabricants sur l'accélération des algorithmes itératifs de reconstruction sur GPU pour réduire la dose en TDM et en CBCT, entre autres Philips pour sons système hybride BrightView XCT de tomographie d'émission monophotomique et de tomodensitométrie. Nous croyons que ce retard est dû tout simplement au long processus de développement, de validation clinique et de certification des applications médicales.

5.8 Conclusion

Les GPUs offrent une grande puissance de calcul qui est moins onéreuse en comparaison aux autres platesformes de calcul parallèle telles que les grappes de CPUs, les ASICs el les FPGAs. Leur utilisation dans la reconstruction de l'images médicales permettrait l'utilisation en clinique des algorithmes de reconstruction complexes, plus performants et exigeant des doses moins

élevées pour les modalités utilisant la radiation ionisante, mais qui sont plus longs à exécuter sur ordinateurs CPUs par rapport aux besoins cliniques. Ces dispositifs permettraient aussi de faire de l'imagerie temps réel avec une meilleure qualité pour diminuer le degré d'invasivité de la radiologie interventionnelle et de la radiothérapie.

Cependant, malgré le développement de la plateforme CUDA qui a facilité la programmation des GPUs, le développement des programmes efficaces en temps de calcul sur GPUs est encore complexe et fastidieux. Il nécessite de bonnes connaissances de l'architecture et des capacités de calcul du GPU utilisé et de la maîtrise des techniques d'optimisation. Les performances de calcul se dégradent rapidement si les différentes règles d'optimisations ne sont pas respectées. La compagnie Nvidia développe en continu des librairies GPUs optimales pour faciliter le développement des applications GPUs performantes. Elle continue aussi à fournir aux utilisateurs des outils qui aident à optimiser et à déboguer les programmes GPUs plus facilement.

CHAPITRE 6

LES LOGICIELS GATE ET STIR

Les simulations des données TEP ont été faites sur le logiciel libre utilisation GATE [91] qui permet de modéliser les systèmes TEP, TEMP et TDM . Nous avons aussi utilisé des fonctions de STIR [218] qui un logiciel libre spécialisé dans la reconstruction des images TEP. Donc nous présentons dans ce chapitre un aperçu de ces différents outils.

6.1 Modélisation Monte Carlo

Les données d'acquisition TEP ont été générées par simulations en utilisant le logiciel GATE développé sur la plateforme GEANT4 de modélisation Monte Carlo [12, 91]. Les calculs ont été effectués sur la grappe de calcul Cottos de Calcul Québec. Quoique GATE est le le plus utilisé pour modéliser les systèmes TEP, d'autres platesformes existent pour réaliser cette modélisation, entre autres EGS_PET, qui est une application basée sur le code Monte Carlo EGSnrc [253], et l'application PENELO_PET [56] basée sur le code Monte Carlo PENELOPE.

6.1.1 GEANT4

Geant4 (*GEometry ANd Tracking*) une plateforme logicielle qui modélise par Monte Carlo le transport des différentes particules dans la matière, a été développé par le Conseil Européen pour la Recherche Nucléaire (CERN). Elle est la version orientée objet développée en C++ de la version précédente GEANT3 écrite en FORTRAN. Elle offre les outils et les librairies pour :

- définir la géométrie et les propriétés physiques du système modélisé ;
- transporter les différentes particules dans la matière ;
- suivre et enregistrer les détails des différentes interactions des particules avec la matière ;
- générer et suivre les événements ;
- visualiser la géométrie du système et les trajectoires des particules en utilisant différents systèmes graphiques tels que OpenGL, Qt et OpenInventor.

Le code source GEANT4, ainsi que de la documentation et des exemples sont en libre accès sur le site de la collaboration GEANT4 (http ://geant4.cern.ch/) [1] est utilisé dans plusieurs

1. Site visité en mai 2015

domaines comme la physique des hautes énergies, la physique médicale et l'astrophysique. Les articles [3, 7] présente plus de détails sur GEANT4.

6.1.2 GATE

6.1.2.1 Architecture et principe de modélisation

GATE est le fruit de collaboration de plusieurs chercheurs pour créer un outil de simulation Monte Carlo en médecine nucléaire simple à utiliser [89, 91] (http ://www.opengatecollaboration.org/) [2]. Ce logiciel est une application orientée objets développée en C++ et organisée en 3 couches au dessus de GEANT4 (figure 6.1) :

1. la couche de base (*core layer*) qui se situe au dessus de GEANT4 et qui implante les classes de base propres à GATE qui concernent la géométrie, les sources radioactives, la gestion du temps, l'analyse des données de la simulation et les classes d'entrées et de sorties ;
2. la couche de développement (*application layer*) qui est un ensemble de classes dérivées des classes des couches inférieures pour définir et spécifier les volumes, les mouvements appliqués aux volumes et sources, les sources, l'analyse des données, les entrées/sorties et les paramètres pour le transport des particules dans la matière ;
3. les scripts utilisateurs (*user layer*) qui offrent à l'utilisateur une interface permettant l'accès aux classes d'application via des scripts pour créer la simulation. L'utilisateur peut soit exécuter en mode interactif les scripts l'un après l'autre, soit créer un fichier ASCII contenant les scripts qu'il exécute sous forme de macro script.

La simulation d'un système TEP par GATE nécessite de décrire : 1) la géométrie du système d'acquisition, 2) la géométrie du fantôme, 3) la chaîne de numérisation qui permet de simuler les différentes transformations que la chaîne électronique de détection applique au signal généré par l'énergie déposée dans les détecteurs par le photon d'annihilation, 4) la physique du transport des particules, 5) le format du fichier des données de sortie, 6) les sources radioactives, 7) le début et la fin de la simulation et 8) le niveau de verbosité.

6.1.3 Système TEP Gemini GXL

Dans notre travail, nous avons modélisé le système d'acquisition Philips Gemini GXL en se basant essentiellement sur l'article de Lamare et al. [106]. Géométriquement, Gemini GXL

2. Site visité en mai 2015

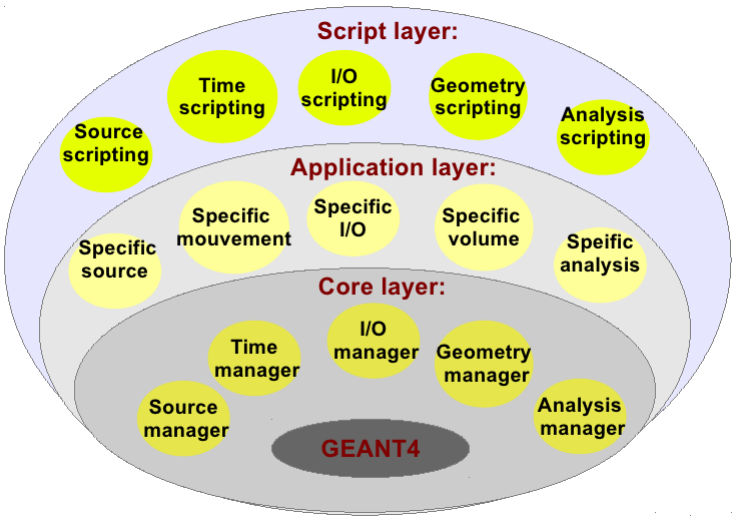


Figure 6.1 – Structure de GATE. Source : Maigne et Perrot [124]

.

est un système qui est composé de 28 blocs de GSO en anneau. Chaque bloc est constitué de 22 détecteurs dans la direction tangentielle et de 29 détecteurs dans la direction axiale, ce qui correspond à 22 anneaux et à 616 cristaux par anneau. La taille de chaque cristal est de 4x6x20 mm^3 respectivement dans les directions tangentielle, axiale et radiale. Chaque bloc peut être en coïncidence avec les 15 blocs lui faisant face. Le nombre de LORs possible est de 85 479 240 (308 nombres de vues x 330 nombres de LORs par vue x 841 plans transversaux). Les détecteurs sont positionnés à une distance radiale de 86.4 cm du centre en permettant un champ de vue de 18 cm dans la direction axiale et 56 cm dans la direction radiale. Pour éliminer les photons provenant de l'extérieur du FOV, deux anneaux en plomb d'épaisseur 2.86 cm protègent les deux anneaux latéraux.

Le modèle Monte Carlo de Gemini mis en oeuvre simule au niveau de la chaîne de numérisation :

- la réponse en énergie du système de détection en introduisant un flou d'énergie gaussien qui correspond à une résolution en énergie de 26% à 511 keV ;
- la fenêtre de sélection d'énergie ayant une largeur de 250 à 750 keV ;
- un temps mort non paralysant de 80 ns au niveau des blocs de détection associé au temps nécessaire pour collecter les photons lumineux créés par le dépôt d'énergie par le photon gamma dans le scintillateur ;

– la fenêtre de temps de coïncidence ayant une largeur 7.5 ns (2τ) pour générer les coïncidences ;
– un temps mort non paralysant de 80 ns associé au temps de traitement au niveau du circuit des coïncidences et du transfert des données vers le disque.

6.1.4 Format de sortie LMF

GATE supporte les formats de données standards ASCII et ROOT, et les formats spécifiques aux systèmes : LMF (*list mode format*), Sinogramme, Ecat7 et Interfile. Nous avons utilisé le format LMF qui est un format développé par le groupe Crystal Clear Collaboration [34]. Il permet d'enregistrer les données en mode liste, c'est-à-dire d'enregistrer les événements en ordre chronologique.

Le format LMF est composé d'un fichier entête sous forme ASCII et un fichier binaire. Le fichier entête qui a une extension .cch indique les données de l'examen et les champs qui décrivent la géométrie du système d'acquisition. Le fichier binaire qui a pour extension .ccs contient les événements simples ou en coïncidence d'une simulation. Ce fichier commence par une entête de longueur variable qui définit l'encodage utilisé pour coder les événements et est suivi par des enregistrements des événements de taille fixe. Ces enregistrements sont de types événements simples ou en coïncidence (*event records*), ou/et taux de comptage (*count rate records*) ou/et informations de simulation GATE (*gate digit record*). Le tableau 6.I présente les principaux champs de ces différents enregistrements.

6.1.5 L'ordinateur utilisé pour les simulations

Le calcul de simulation a été effectué sur la grappe de calcul Cottos de Calcul Québec. Cet ordinateur qui est localisé à l'université de Montréal est composé de 128 nœuds de calcul. Chaque nœud est constitué de deux processeurs Intel Xeon E5472 quadri-cœurs de fréquence 3 GHz et possède 16 Go de mémoire RAM.

Un script a été développé pour exécuter chaque simulation sur 32 processeurs. Pour ce faire, 32 simulations indépendantes sont lancées simultanément sur 32 processeurs. Pour ne pas modifier les paramètres physiques tels que le temps morts, le taux de comptage, le taux de diffusé et des coïncidences aléatoires, chaque simulation n porte sur l'intervalle de simulation $[t_{n-1}, t_n]$. Pour plus d'efficacité de calcul, la valeur de chaque t_n a été calculée selon la formule proposée

Tableau 6.I – Les principaux champs des enregistrements LMF d'un évènement simple

Champ	Taille en octets	Système réel	Simulation Gate
Évènement			
Temps de détection	8	Oui	Oui
Énergie déposée	1	Oui	Oui
Identité du détecteur	2	Oui	Oui
Position angulaire du système TEP	2	Oui	Oui
Position axiale du système TEP	2	Oui	Oui
Taux de comptage			
Taux de comptage des évènements	2	Oui	Oui
Taux de comptage des coïncidences	2	Oui	Oui
Taux de comptage des coïncidences aléatoires	2	Oui, retardées	Oui
Simulation Gate			
Identité de la simulation	4	Non	Oui
Identité de l'événement	4	Non	Oui
Identité de la source	2	Non	Oui
Nombre de Compton dans le fantôme	1	Non	Oui
Nombre de Compton dans le détecteur	1	Non	Oui

par De Beenhouwer et al. [48] comme suit :

$$t_n = \frac{ln((N-n)/n)e^{-\lambda t_s} + n/Ne^{-\lambda t_s}}{-\lambda}, \tag{6.1}$$

où t_s est la durée totale de simulation, λ est la constante demi-vie du traceur utilisé et N est le nombre processeurs de calcul utilisés, qui est de 32 dans notre cas.

La combinaison des 32 fichiers LMF créés par les 32 simulations parallèles a été réalisée par un programme écrit en utilisant la librairie LMF_v3.0.

6.2 Logiciel STIR

STIR est un logiciel écrit en C++ et destiné à la reconstruction et au traitement des images en TEP [218, 219] (http ://stir.sourceforge.net/main.htm) [3]. C'est un ensemble de librairies contenant des classes, des fonctions et des utilitaires qui portent sur la reconstruction 3D PET. Ce logiciel a été originalement développé dans le cadre du projet PARAPET (1997–1999) de l'Union Européenne pour le développement des algorithmes de reconstruction 3D en PET. Il a été publié comme un code source ouvert en juin 2000. Dans notre travail, nous avons implanté

3. Site visité en mai 2015

sur CPU, en utilisant les librairies de STIR, les algorithmes de reconstruction à partir des sinogrammes MLEM et OSEM et ceux de reconstruction à partir du mode liste LM-EM et LM-SEM. Pour les implantations sur GPU, nous avons aussi utilisé des fonctions STIR, notamment des fonctions de la librairie *InputOutput* pour lire et écrire les images, les classes *Proj data* pour lire les données en sinogrammes, et celles de la librairie *listmode_buildblock* pour lire les données en mode liste sous le format LMF. Nous avons aussi modifié quelques fonctions de cette librairie pour les rendre plus fonctionnelles.

6.2.1 L'architecture de STIR

STIR est un code orienté objet qui est très hiérarchisé et qui utilise toutes les fonctionnalités qu'offre C++, notamment l'héritage, les classe abstraites, le polymorphisme, et les pointeurs intelligents pour optimiser l'implémentation. Cette hiérarchisation poussée a permis d'optimiser le code en évitant la redondance dans le codage des fonctions qui sont communes à des classes différentes et facilite son extension par le développement de nouvelles fonctionnalités. Elle permet également d'utiliser les classes dérivées via des classes supérieures d'interface qui cachent les détails de l'implémentation à l'utilisateur. STIR utilise aussi les templates C++ permettant d'écrire un code générique pour des données de différentes natures. Un autre avantage important de STIR est qu'il permet de spécifier certaines fonctionnalités durant son exécution, entre autres le format des données d'entrée et de sortie et le format des projections. Cependant, l'utilisation de STIR est complexe et nécessite, en plus de la compréhension de la structure de STIR, une solide connaissance en C++.

Par ailleurs, le logiciel STIR est organisé en trois parties :

1. Les librairies contenant les classes portant sur la définition de différents systèmes TEP, les tableaux multidimensionnels définissant des structures des données, la projection et la rétro-projection, les symétries, les fonctions objectives, les reconstructions analytiques et itératives, la correction des données de projection telle que la normalisation, l'atténuation, la géométrie et l'estimation de diffusé, la lecture et l'écriture (input/output) de différents format de données, notamment : sinogrammes, *GE Advance sinogram data*, ECAT6, ECAT7 et interfiles, LMFs, *list-mode ECAT EXACT HR+*, etc., modélisation cinétique, transfert des données aux paramètres durant l'exécution (*parsing data on run-time*), différents filtres 1D, 2D,3D, etc.

2. Les utilitaires développés à partir des classes de librairies pour manipuler, filtrer, et corriger les images et les données des projections.
3. Une plate-forme de test pour s'assurer du bon fonctionnement des classes de différentes librairies lors de l'installation du logiciel.

Troisième partie

Travaux et résultats

CHAPITRE 7

ARTICLE 1 : FAST GPU-BASED COMPUTATION OF THE SENSITIVITY MATRIX FOR A PET LIST-MODE OSEM ALGORITHM

Publié dans le journal Physics in Medicine and Biology (soumis le 10 janvier 2012, révisé le 11 avril 2012 et publié le 14 septembre 2012)

Auteurs : Moulay Ali Nassiri[1], Sami Hissoiny[2], Jean-François Carrier[1] et Philippe Després[3].

[1]*Département de radio-oncologie, Centre hospitalier de l'Université de Montréal (CHUM), Montréal (Québec), CANADA.*

[2]*Département de génie informatique et génie logiciel, École polytechnique de Montréal, Montréal (Québec), CANADA.*

[3]*Département de radio-oncologie, Centre hospitalier universitaire de Québec (CHUQ), Québec (Québec), CANADA et Département de physique, de génie physique et d'optique, Université Laval, Québec, CANADA.*

Contribution des auteurs :

Moulay Ali Nassiri est le principal auteur de cet article ; il a apporté la plus grande contribution à ce travail en :

- effectuant la recherche bibliographique sur la reconstruction des images TEP à partir des données d'acquisition en mode liste ;
- déterminant la problématique que constitue le calcul de la matrice de sensibilité dans l'utilisation en clinique quotidienne des algorithmes itératifs stochastiques pour la reconstruction des images TEP à partir des données en mode liste, et en formulant l'objectif du travail ;
- modélisant par Monte Carlo le système d'acquisition TEP de Philips Gemini GXL ;
- parallélisant l'algorithme LM OSEM et en développant celui qui calcule la matrice de sensibilité
- implémentant et optimisant sur GPU l'algorithme LM OSEM et celui qui calcule la matrice de sensibilité ;

– analysant les résultats et rédigeant le manuscrit publié.

Jean-François Carrier et Philippe Després ont encadré activement le travail durant ses différentes phases. Ils ont aussi restructuré et corrigé le manuscrit initial et final, et ils ont aidé l'auteur principal à bien formuler les réponses aux commentaires des arbitres.

Sami Hissoiny a initié l'auteur principal sur la programmation des GPUs avec CUDA et a corrigé le manuscrit initial.

7.1 Résumé et mise en contexte

– **Problématique et objectif** : la reconstruction à partir des données en mode liste présente plusieurs avantages par rapport à la reconstruction à partir des données organisées en sinogrammes. Ces avantages sont entre autres de corriger avec plus d'efficacité lu flou du mouvement et d'utiliser avec plus de précision du temps de vol pour améliorer la quantification [183, 186, 190]. Le mode liste est aussi plus rapide et plus efficace pour les acquisitions dynamiques où le nombre d'événements est inférieur au nombres des LORs. Il permet la reconstruction des images 4D et des paramètres physiologiques avec une bonne résolution temporelle et un bon RSB [193]. La problématique est que le temps de reconstruction à partir des données en mode liste est relativement long pour une utilisation clinique de routine.

  Pour remédier à ce problème du temps de calcul, plusieurs travaux d'accélération sur GPU de l'algorithme LM-OSEM ont été effectués [35, 112, 127, 186, 189, 234]. Cet algorithme est une méthode de reconstruction stochastique à partir du mode liste et dont la version de reconstruction à partir des sinogrammes (OSEM) est la plus utilisée en clinique. Cependant, ces travaux ont omis le fait que le calcul de la matrice de sensibilité peut être plus long que l'exécution de l'algorithme LM-OSEM pour les TEPs modernes [35, 127, 175]. Ils supposent que cette matrice peut être précalculée et stockée pour une utilisation future. Le problème est qu'elle doit être calculée pour chaque patient à partir de sa propre matrice des coefficients d'atténuation obtenue d'une acquisition tomodensitométrique lors de la réalisation de l'examen TEP. De plus, cette matrice doit être calculée pour chaque fenêtre pour les acquisitions dynamiques 3D synchronisées. Par conséquent, le gain en temps d'exécution obtenu par l'accélération de LM-OSEM sur GPU sera annulé par le temps nécessaire à l'estimation de la matrice de sensibilité. Ce travail a pour objectif d'ac-

célérer sur GPU l'exécution l'algorithme stochastique 3D LM-OSEM incluant le calcul de la matrice de sensibilité qui, afin d'introduire la reconstruction 3D à partir du mode liste en clinique de routine.

– **Méthodologie** : la mise en oeuvre de l'algorithme a été effectuée sur le GPU Tesla C2050 pour le système Philips Gemini GXL qui a 85 millions de LOR. La matrice système est calculée en temps réel par la méthode multi-trajectoires de Siddon en utilisant 6 trajectoires par LOR : 3 lignes dans la direction tangentielle et 2 lignes dans la direction axiale, afin d'améliorer l'échantillonnage et d'augmenter, par conséquent, l'exactitude de la quantification. Par ailleurs, pour accélérer le calcul de la matrice de sensibilité, nous avons exploité les 8 symétries de base dans les plans transversaux, et nous avons réarrangé les données des matrices d'atténuation, de sensibilité et de normalisation pour assurer la coalescence d'accès à la mémoire global de GPU. L'implémentation permet la reconstruction des images pour les deux définitions utilisées en clinique par Philips, qui sont les images de 188 x188x57 voxels de taille 2x2x3.15 mm^3 et celles de 144x144x57 voxels de taille 4x4x3.15 mm^3.

– **Résultats et discussion** : notre mise en oeuvre permet de calculer en 9 secondes la matrice de sensibilité de définition 188x188x57 et en 8 secondes et celle de définition 144x144x57. Les résultats obtenus démontrent aussi que l'utilisation de plus de symétries que les 8 symétries de base dans le plan axial ne permettent pas d'accélérer la reconstruction de la matrice de sensibilité. En effet, l'exploitation de plus de symétries engendre l'utilisation de plus de registres par les SMPs de GPU, et, par conséquent, diminue leur occupation. Nous avons aussi montré que le fait de ne pas utiliser la fonction *atomicAdd()* a un impact minime sur la précision de la quantification, alors que son utilisation pénalise énormément l'efficacité de calcul en multipliant par 6 le temps de calcul.

  Concernant l'algorithme LM-OSEM, les temps de calcul obtenus sont de 0.8 et 1.1 seconde par million d'événements par itération pour respectivement des matrices de taille 144x144 x 57 voxels et 188 x188x57 voxels. Quoiqu'ils soient comparables aux résultats obtenus par d'autres auteurs, le temps par événement pour exécuter LM-OSEM est 9 fois supérieur à celui par LOR pour le calcul de la matrice de sensibilité. Ceci s'explique par le fait que le mode liste ne permet pas d'utiliser les symétries pour accélérer la reconstruction et que l'accès aux différentes matrices ne peut être optimisé pour respecter les conditions de coalescence. Les opérations de lecture et d'écriture respectivement de la matrice im-

age et dans la matrice gradient se font de manière aléatoire, car les adresses ne sont pas déterminées avant l'exécution du *Kernel*.

– **Conclusion** : dans ce travail, nous avons pu atteindre notre objectif qui est de calculer en moins de 10 secondes, pour le système de Philips Gemini GXL, la matrice de sensibilité intégrant les facteurs d'atténuation du patient et les coefficients de normalisation des détecteurs. Ce temps permettra non seulement d'utiliser la reconstruction 3D à partir du mode liste en clinique de routine, mais il permettra aussi d'envisager des applications plus avancées de la reconstruction à partir du mode liste telles que la reconstruction dynamique en temps réel et la reconstruction 4D. Pour atteindre cet objectif, il faut dépasser les obstacles, cités ci dessus, qui entravent l'accélération sur GPU de ce type de reconstruction. Par ailleurs, notre implémentation nécessite l'intégration des fonctions de correction du diffusé et des événements aléatoires dont le temps d'exécution est relativement long dans le cas du mode liste, et il nécessite aussi d'être validée en utilisant des données cliniques.

7.2 Abstract

During the last decade, studies have shown that 3D list-mode ordered-subset expectation-maximization (LM-OSEM) algorithms for Positron Emission Tomography (PET) reconstruction could be effectively computed and considerably accelerated by Graphics Processing Units (GPUs) devices. However, most of these studies rely on pre-calculated sensitivity matrices. In many cases, the time required to compute this matrix can be longer than the reconstruction time itself. In fact, the relatively long time required for the calculation of the patient-specific sensitivity matrix is considered as one of the main obstacle in introducing a list-mode PET reconstruction algorithm for routine clinical use.

The objective of this work is to accelerate a fully 3D LM-OSEM algorithm, including the calculation of the sensitivity matrix that accounts for the patient-specific attenuation and normalisation corrections.

For this purpose, sensitivity matrix calculations and list-mode OSEM reconstructions were implemented on GPUs, using the geometry of a commercial PET system. The system matrices were built on-the-fly by using an approach with multiple rays per detector pair. The reconstructions were performed for a volume of 188x188x57 voxels of 2x2x3.15 mm^3 and for another volume of 144x144x57 voxels of 4x4x3.15 mm^3.

The time to compute the sensitivity matrix for the 188x188x57 array was 9 seconds while the LM-OSEM algorithm performed at a rate of 1.1 millions of events per second. For the 144x144x57 array, the respective numbers are 8 seconds for the sensitivity matrix and 0.8 million of events per second for the LM-OSEM step. This work lets envision fast reconstructions for advanced PET applications such as real time dynamic studies and parametric image reconstructions.

7.3 Introduction

Two modes of data acquisition can be used in Positron Emission Tomography (PET), namely the conventional histogram-mode and the list-mode. In histogram-mode, the coincidence events are organized in sinogram bins (projections) and are often compressed in order to accelerate the image reconstruction and to reduce the sinogram size. In list-mode acquisitions, the detected coincidence events are stored sequentially, event by event as they are acquired, in a long list file. The attributes stored for each coincidence are typically the detector pair ID, the time of detection and the energy deposited in each detector.

For modern PET scanners, many studies have established that list-mode reconstructions can be more efficient than histogram-mode algorithms [183, 186, 190]. This is especially true for dynamic PET studies in which the number of events acquired in each 3D frame is typically less than the number of bins in a full sinogram set. The reconstruction from list-mode data offers other advantages over the histogram-mode, such as motion correction, the possibility of using time-of-flight (TOF) information to improve the quantification accuracy, and the possibility of using temporal basis functions in 4D image reconstructions [193]. In this context, the list-mode expectation-maximization algorithm (LMEM), expanded from the maximum-likelihood expectation-maximization (MLEM), and its accelerated version list-mode ordered-subset expectation-maximization (LM-OSEM) algorithm are worth investigating [35, 112, 127, 186, 189, 234].

One of the main obstacle in introducing the list-mode reconstruction approach for routine clinical use is the relatively long time required to compute the sensitivity matrix [35, 127, 175]. Modern PET scanners have a large number of lines of response (LOR) and the exact calculation of the sensitivity matrix uses all possible LORs passing through the object, leading to calculation times that could be longer than the reconstruction time itself. Matej *et al.* for instance reported

calculation times on the order of hours for this kind of task [127]). Furthermore, the sensitivity matrix must be calculated for each patient based on its own attenuation map obtained from a transmission scan [35, 66, 127, 175, 184]. To accelerate the calculation of the sensitivity matrix, variance reduction techniques based for example on down-sampling the number of LORs have been used [35, 175]. These approaches however add errors to the sensitivity matrix and thus potentially degrade the quality of the reconstruction [175, 177].

Previous studies aimed at accelerating list-mode reconstructions either on CPU [189, 202, 234] or GPU devices [57, 170–172, 198] have assumed a pre-computed sensitivity matrix. On the CPU, one approach consists in using the SIMD instructions set [77]. Although this strategy typically yields interesting results, the architecture of modern GPUs and the associated programming model are becoming very attractive for high-performance computing.

In order to perform all steps required for advanced PET reconstruction, a 3D LM-OSEM algorithm that integrates the computation of the sensitivity matrix was implemented on a NVIDIA Tesla C2050 GPU. The main objective of this work was to verify if this workload could be achieved within clinically compatible time frames. This would open the way to advanced list-mode acquisitions in the clinic with potential benefits regarding uptake quantification and pharmacokinetic modeling. Ultimately, significant kinetic parameters could be extracted from list-mode acquisitions, with potential benefits for the sensitivity and specificity of diagnostic PET exams. The monitoring of the therapeutic response with PET would also benefit from list-mode acquisitions, in which the accuracy of uptake quantification can be improved.

For this study, the reconstructions were performed for simulation data modeling a Philips Gemini GXL PET scanner. Corrections for random and scatter events were not implemented at this time.

7.3.1 List-Mode Ordered-Subset Expectation-Maximization (LM-OSEM)

To estimate the activity of radiotracer in an object parametrized by a set of voxels $M = \{1,2,...,J\}$, the LM-OSEM algorithm [162] splits the list-mode events space S into L roughly equal sized disjoint data subsets, $\{S_1, S_2, ..., S_L\}$, and processes the image at each sub-iteration according to :

$$\lambda_j^m = \frac{\lambda_j^{m-1}}{N_j} \sum_{k \in S_l} p_{i_k,j} \frac{1}{\sum_{b=1}^{J} p_{i_k,b} \lambda_j^{m,l-1} + \frac{s_{i_k}+r_{i_k}}{a_{i_k}\varepsilon_{i_k}}} \tag{7.1}$$

$$N_j = \sum_{i=1}^{N} p_{i,j} a_i \varepsilon_i, \tag{7.2}$$

where λ_j^m is the intensity value in voxel j estimated after m sub-iterations and S_l is the subset data for $l = m \bmod L$. The coefficients $p_{i,j}$ are the elements of the geometric system matrix (SM) defined as the probability that a photon pair produced in voxel j reaches the front faces of the detector pair i in the absence of attenuation and assuming perfect photon-pair co-linearity [180]. N_j is the sensitivity matrix that accounts for sensitivity variations due to attenuation and normalization. Finally, s_{i_k} and r_{i_k} are respectively the expected mean scatter counts and expected mean random counts along the i^{th} LOR corresponding to the k^{th} coincidence in the list-mode file.

The SM is extremely large, especially for 3D modern scanners. Despite the fact that this matrix is sparse and presents symmetries that can be exploited to reduce its size, on-the-fly computation of the $p_{i,j}$ coefficients is more efficient than a pre-calculated matrix [193], mainly for the computation of LM-OSEM on GPU. In fact, the list-mode data is not arranged in any regular pattern, so reading pre-calculated system coefficients from GPU global memory is a random access process that can not be optimized neither by using texture or cached global memory. Using on-the-fly computation of SM also allows to free more memory space to store event streams and to optimize the transfer of data from the CPU memory to the GPU memory. The SM are commonly computed on-the-fly by using a simple raytracing algorithm such as the one proposed by Siddon [209], where $p_{i,j}$ become radiological path lengths (RPLs), *i.e.* the length of intersection of LOR i and voxel j multiplied by the attenuation coefficient of this specific voxel. To provide a better sampling of the image volume, a multiple rays per detector pair approach was used in this work [139].

The sensitivity matrix incorporates the attenuation and normalization corrections as weight factors to provide more accurate modelling of the measurement process and consequently lead to a better noise-resolution trade-off [96, 136, 184]. The computation of this matrix requires for each LOR i 1) the estimation of the attenuation correction factors (ACFs) a_i by forward-projection of the PET attenuation coefficient map M_{μ_j} obtained from the transmission scan according to $a_i = exp(-l_j \sum_{j=1}^{j=J} p_{ij}\mu_j)$, where l_j is the length of voxel, and 2) the back-projection of

the sensitivity factors $w_i = \varepsilon_i \times a_i$ where ε_i is the normalisation factor (NF). The NF accounts for sensitivity variations due to non-uniformities in the detector efficiencies and to geometric factors. Since modern 3D scanners have a huge number of LORs (order of 10^8), the computation time of the sensitivity matrix could be as long as the reconstruction time itself [35, 175].

7.3.2 Multiple rays per detector pair

The LOR joining the center of two detectors is generally described with four parameters : ϕ, s, z and θ, where ϕ and s are the azimuthal angle (view) and tangential position in the transverse plane, z is the axial position and θ is the LOR tilt relative to the perpendicular transaxial plane. A set of LORs with a common z and θ defines an histogram and a set of histogram with a common θ defines a segment.

The multiple rays per detector pair approach consists in virtually dividing each detector into m sub-detectors in the tangential direction and n sub-detectors in the axial direction, leading to sub-LORs. The coordinates of these sub-LORs are calculated according to a virtual scanner geometry where the number of detectors per ring is multiplied by m and the number of rings is multiplied by n. During the back-projection step of the reconstruction algorithm, the projection values of each detector pair are divided equally over its sub-detector pairs. For the forward-projection step, the calculated projection values for the related sub-detector pairs are added together and normalised by $m \times n$ to calculate the projection value of the corresponding detector pair.

The number of sub-LORs used per detector pair leads to a trade-off between accuracy and calculation time of the SM. In this work, the optimal number of sub-LORs was determined empirically, as explained in section 7.4.6 below.

7.3.3 Symmetries

In a cylindrical PET scanner, many geometrical symmetries are present and could be used to accelerate the computation of the SM [253]. For the grid array, the scanner presents three view symmetries : $90^\circ - \phi$, $90^\circ + \phi$, $180^\circ - \phi$ and one reflection symmetry (swap s) in the transverse plane (figure 7.1). If the z voxel size chosen is a integer fraction of the distance between the center of two nearby rings (axial pitch), axial translational symmetries and axial reflection symmetry (swap segment) could be used to decrease the computation time of the SM [87].

7.4 Methods

7.4.1 GPU implementation

In this work, a NVIDIA Tesla C2050 GPU was used. This device has 448 cores and 3 GB of memory with 144 GB/s of bandwidth. Since it supports CUDA compute capability 2.0, the global memory is cached up to 48 kB per multiprocessor and supports the floating-point atomic addition operating on 32-bit words in global and shared memory. Atomic addition guarantees that only a single thread has access to a piece of memory during the execution of addition. This feature is very important in PET image reconstruction since it allows the reduction of race conditions resulting in data loss in the calculation of the back projection due to a simultaneous read-modify-write process of parallel threads. The race conditions occurs when two or more threads access the same memory element at the same time to read/modify it. For compilation, `gcc` version 4.4.1 for the C++ code and CUDA version 3.2 for the GPU code were used.

7.4.2 Computation of the sensitivity matrix

The GPU implementation of the sensitivity matrix calculations consisted essentially in associating each sub-LOR to an execution thread. The attenuation coefficient map was stored in a 3D texture while the normalisation factor matrix used CUDA linear memory in global memory. This strategy allows for an efficient use of texture cache and global memory cache. The CUDA atomic float operation `atomicAdd()` was used to update values in the sensitivity matrix stored as linear memory. Reducing the number of global memory accesses was an important objective of this work. Efforts were made to increase the global memory throughput by mapping these matrices so that coalesced memory accesses are possible.

To exploit the symmetries of the Gemini GXL system, symmetrical LORs are grouped together to form disjoint LORs subset and each thread was related to a basis LOR of one subset. The attenuation coefficient matrix was forward-projected along each basis LOR and along its symmetric LORs using multiple rays per detector pair. The resulting attenuation correction factors were stored in shared memory as a vector $\mathbf{W}$ that was mapped to fulfill the coalesced access requirements. The system matrix coefficients p_{ij} are calculated on-the-fly : for every sub-ray i related to the basis LOR associated to thread L, the algorithm determines p_{ij} in each voxel V_j encountered as the radiological path length RPL_{ij}. This value is multiplied by the attenuation coefficient of voxel V_j and the result is added to $\mathbf{W}[L]$ to calculate the attenuation correction

factor corresponding to the basis LOR. For each symmetry m used, the voxel $Vsym_j$ symmetric to voxel V_j is determined and the corresponding value is added to $\mathbf{W}$[L+m*blockDim] were blockDim is a GPU execution parameter.

For the basis LOR connected to the thread and for its symmetric LORs LOR_{sym}, the attenuation correction factors (ACF) and normalization factors (NF) are combined and stored in $\mathbf{W}$. Finally this vector is back-projected in the sensitivity matrix using multiple rays per detector pair and exploiting symmetries as in forward-projection.

The strategy combining the forward-projection and back-projection in the same thread improves the computation performance by cutting down the access to global memory to store the forward-projection values for all LORs. As the GPU calculation kernels are executed asynchronously, the control is returned to the host thread before the device has completed the computation of the sensitivity matrix. In other words, the CPU host function that reads events from the list-mode file is overlapping with computation of the sensitivity matrix on GPU. When the GPU execution of the sensitivity matrix kernel is finished, all the matrices are deleted from device memory except for the sensitivity matrix which will be used in the LM-OSEM algorithm.

7.4.3 GPU memory mapping strategy

The accesses to the sensitivity matrix and to the attenuation coefficient matrix are random over threads. In order to increase the memory throughput, these matrices were reordered so that the symmetrical voxels in-plane (figure 7.1) become neighbours. The attenuation matrix which is a read-only matrix is mapped as a 3D CUDA array with type float4 and is associated to a 3D texture. Each element then stores the attenuation coefficients of four symmetrical voxels corresponding to the four symmetrical views in-plane (figure 7.2). The sensitivity matrix, which is a read-write matrix accessed using atomic operations, is mapped as a CUDA linear memory of type float8. Each element of this data structure stores the sensitivity values for the eight symmetrical voxels corresponding to the in-plane view symmetries and reflection symmetry.

The GPU global memory has a coalescing behaviour which means that simultaneous global memory accesses by threads within a warp can be serviced by one or many transactions according to the size of the word accessed by each thread and the mapping of the memory addresses of these words. For devices of CUDA compute capability 2.x, the global memory bandwidth is used most efficiently when the simultaneous memory accesses by threads in a warp is coalesced into a single memory transaction of 128 bytes. This optimal coalesced access to global memory is achieved if

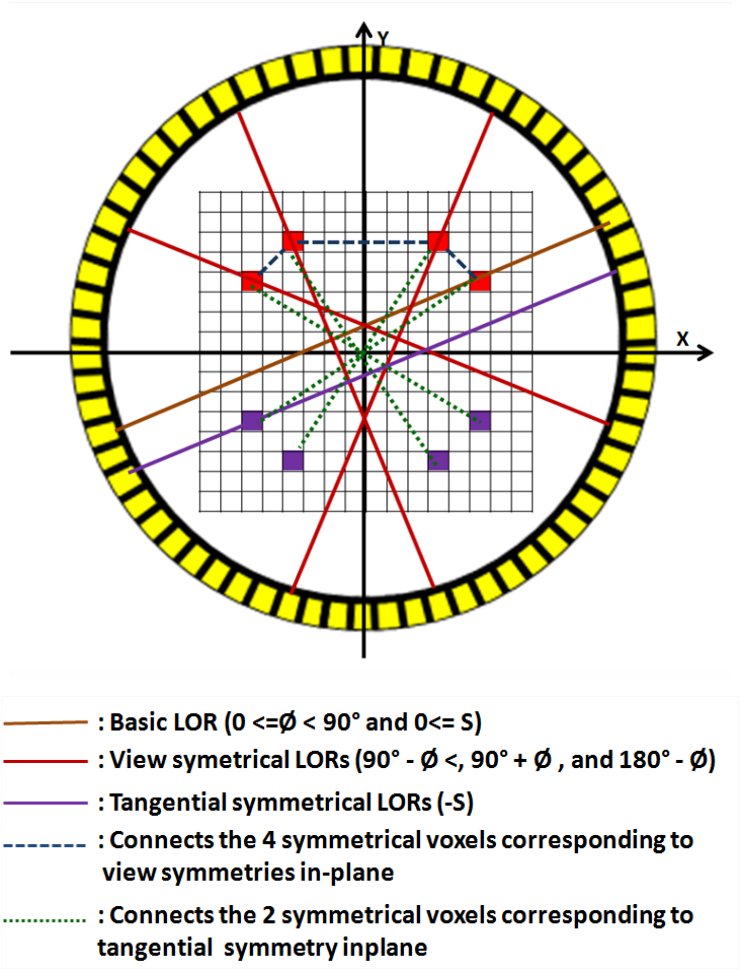


Figure 7.1 – In-plane symmetries used and their corresponding symmetrical voxels.

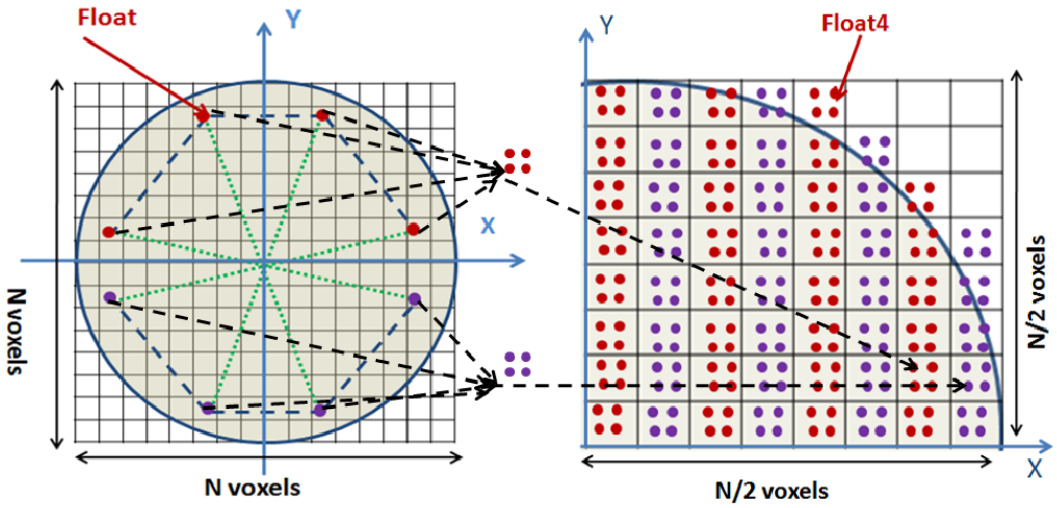


Figure 7.2 – Mapping of attenuation coefficients matrix on the GPU memory.

specific requirements are met [151]. To meet these requirements, the NF matrix was mapped on GPU global memory as CUDA linear memory of type `float4`. Each element of the array stores the values of normalisation factors corresponding to symmetrical views (where $V = 308$ is the total number of views for the Gemini GXL PET scanner) and the data is reoredred as shown in figure 7.3.

For each segment, the threads corresponding to the basis LORs read data row by row. Each row stores $V/4$ x $S/2$ `float4` values, where V is the total number of views and S is the total number of tangential LORs in-plane for each view ($V = 308$ and $S = 330$ for the Gemini GXL). Since optimal coalescing is achieved when the size of each row is a multiple of 128 bytes, their size was increased to meet this criteria (28 `float4` elements added for the Gemini GXL).

7.4.4 Implementation of the LM-OSEM algorithm

Two GPU execution kernels were used in the implementation of the LM-OSEM algorithm. The first one is event-based and computes the gradient of the image at each iteration and for each subset event (frame) while the second updates the image. Each thread associated to one event forward-projects the image along the LOR related to the event, and back-projects immediately the inverse of projection values in the gradient matrix using atomic float operations. As for the sensitivity matrix calculations, the strategy consists in computing the forward projection and the

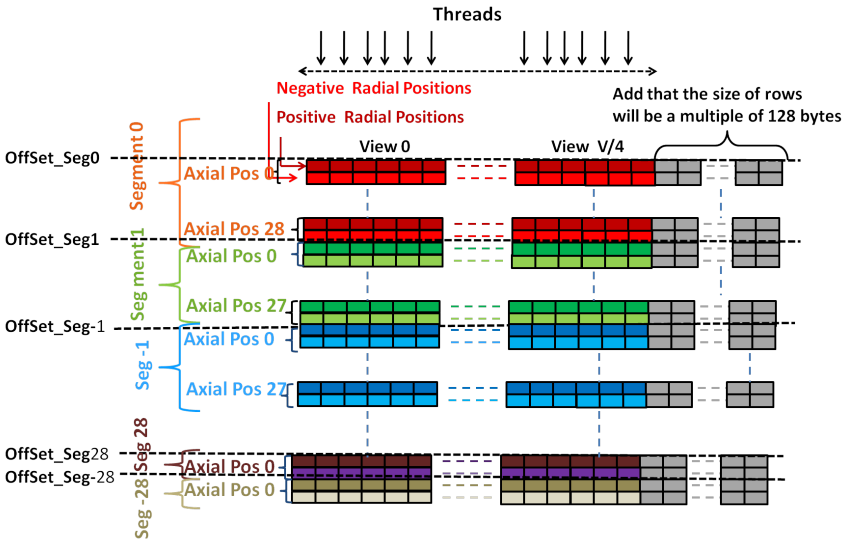


Figure 7.3 – Mapping of the normalisation matrix on the GPU global memory.

back projection in the same thread, avoiding a space allocation in the device global memory for storing the forward projection results of all threads. Therefore, the computation performance was improved by cutting down the access to global memory and freeing more memory space to store the event stream. To avoid race conditions in back-projection, the atomicAdd() function was used. In fact, Cui et al. [45] showed that the efficient strategy to reduce data loss in the back-projection for LM-OSEM algorithm is to use LOR driven approach with hardware atomic operation. The execution kernel updating the image is voxel-based. It multiplies the current image by its gradient and normalises the result using the sensitivity matrix voxel by voxel.

To evaluate the computation efficiency of our GPU implementation, we coded the fully 3D LM-OSEM algorithm that includes the sensitivity matrix computation on CPU using the *ProjMatrixByBinUsingRayTracing* and *BinNormalisationFromAttenuationImage* classes of the *Software for Tomographic Image Reconstruction* (STIR) project [218]. The input/ouput library of STIR was also used to read and write the image arrays in Interfile format.

7.4.5 Monte Carlo simulations and quantitative measurements

7.4.5.1 PET scanner

To generate the normalisation matrix and list-mode data, Monte Carlo simulations were performed using the GATE (Geant4 Application for Tomographic Emission) package [90] to simulation data acquisition on a Philips Gemini GXL PET scanner [106]. The Gemini GXL is composed of 28 blocks, each being a set of 22 tangential and 29 axial individual GSO crystals. The dimension of each crystal is 4, 6 and 20 mm in the tangential, axial and radial directions respectively. This scanner offers a field-of-view (FOV) of 56 cm radially and 18 cm axially. Each block can be in coincidence with any 15 opposite blocks, resulting in $85,479,240$ LORs. Assuming single precision, these LORs require 342 MB of memory for the normalisation matrix.

7.4.5.2 Normalisation factor matrix

A uniform non-attenuating cylindrical phantom (air) with a diameter of 50 cm and a length of 18 cm was used to calculate the normalisation factor matrix using Monte Carlo simulations. The phantom was filled with 5 mCi of ^{18}F. All the possible camera symmetries (14 rotational, 28 translational and one reflection symmetries) were used to increase the number of effective counts in each LORs in order to decrease the statistical noise [253]. Because the detector response was uniform in the Monte Carlo simulations, only the geometric factors were corrected. To get the prompt events, a coincidence sorter was applied using a predefined coincidence time window of 7.5 ns. The data was then stored in a 3D matrix (308 views $\times$ 330 tangential positions $\times$ 841 histograms) and reordered to allow coalesced GPU memory access as explained in section 7.4.3.

7.4.5.3 Contrast recovery coefficient and noise

A cylindrical water phantom (30 cm diameter, 20 cm length, 2.2 mCi of ^{18}F) with four hot spherical lesions of sizes 11, 13, 17, and 22 mm and of contrast 4 : 1 was simulated [127]. The lesions were located in the central slice and positioned radially at 6 cm from the center. The four lesion pattern was repeated three times. List-mode data sets of 114×10^6 coincidences were generated. The simulated data was used to validate the proposed implementation and to evaluate the global image quality by estimating the contrast recovery coefficient (CRC) in the reconstructed images.

The CRC and noise level as defined in the NEMA NU 2007 protocol [147] were used. For the hot spheres the CRC was calculated with

$$CRC_H = \frac{C_H/C_B - 1}{C} \text{x} 100\% \tag{7.3}$$

where C_H is the average count in the region of interest (ROI) for the given hot sphere. C_B is the background count and C is the actual contrast. To determine C_B, eight ROIs were placed in different parts of the background with the same size as the lesion of interest and C_B was obtained by averaging the counts of these eight ROIs over four realisations. As suggested by the NEMA protocol, the background ROIs were drawed 15 mm away from the edge of the phantom and from the hot lesions. The noise for the ROIs was calculated as the ratio between the mean standard deviation S_D over 32 background ROIs and the mean contrast of background C_B.

7.4.6 Reconstructions

The reconstructions were computed for two image arrays. The first one presents a field-of-view (FOV) of 37.6 cm and is a $188 \times 188 \times 57$ image of $2 \times 2 \times 3.15$ mm^3 voxels. The second one covers the entire FOV (57.6 cm) offered by the Gemini system and is a $144 \times 144 \times 57$ image of $4 \times 4 \times 3.15$ mm^3 voxels. The choice of voxel sizes in the axial direction having half the ring pitch (6.3 mm) allows axial symmetries that decrease the computation time.

It was determined empirically that for the $2 \times 2 \times 3.15$ mm^3 voxel sizes, 3 tangential rays and 2 axial rays per detector pair led to optimal results in terms of the relative mean square error (compared to an image obtained with 10 tangential rays) vs calculation time (data not shown). The respective numbers for the $4 \times 4 \times 3.15$ mm^3 voxel sizes were 2 tangential rays and 2 axial rays per detector pair. The calculation time increased proportionally with the number of sub-LORS used.

7.5 Results

7.5.1 Sensitivity matrix

Table 7.I reports the execution times for the sensitivity matrix calculations on the GPU device, obtained with and without reordering the memory for coalescence in the case where the symmetries of the scanner are not exploited. If the attenuation coefficients and sensitivity ma-

trices are not coalesced, using more symmetries does not decrease the computation time of the sensitivity matrix on the GPU. In fact, exploiting symmetries requires the use of GPU shared memory to store the forward-projection values corresponding to all symmetrical LORs associated to a thread or to employ more registers per thread. As a consequence, the occupancy of each multiprocessor becomes lower with strategies exploiting symmetries. Therefore, it seems that the gain on computation time of SM coefficients due to axial symmetries is hindered by the decrease of GPU multiprocessors occupancy.

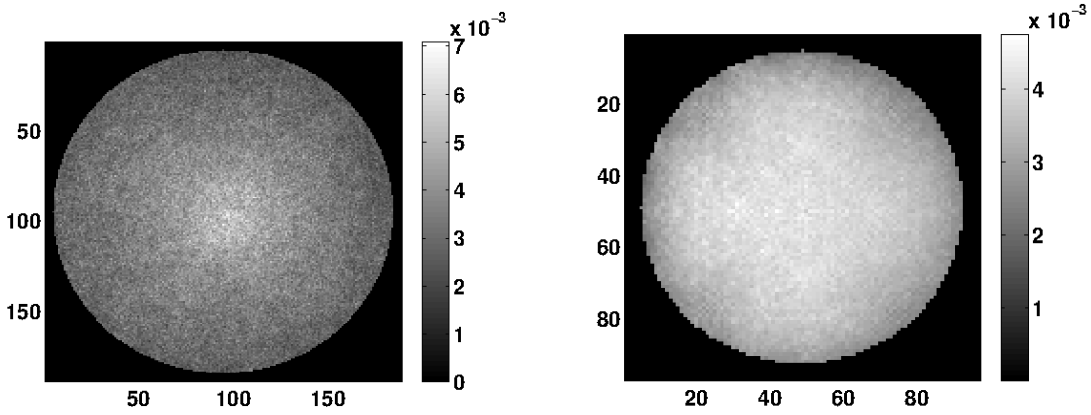
Tableau 7.I – Computation time in seconds for sensitivity matrices on a Tesla C2050 GPU with different strategies.

	$188 \times 188 \times 57$ matrix of $2 \times 2 \times 3.15$ mm^3 (2 axial sub-rays) (3 tangential sub-rays)		$144 \times 144 \times 57$ matrix of $4 \times 4 \times 3.15$ mm^3 (2 axial sub-rays) (2 tangential sub-rays)	
	Symmetries not exploited			
	Non-coalesced	Coalesced	Non-coalesced	Coalesced
Without `atomicAdd()`	52.1		28.6	
With `atomicAdd()`	56.0		47.2	
	Using in-plane symmetries			
Without `atomicAdd()`	62.8	8.3	36.7	7.2
With `atomicAdd()`	69.3	51.6	47.3	49.8
	Using in-plane and axial symmetries			
Without `atomicAdd()`	65.5	10.5	39.9	8.7
With `atomicAdd()`	72.2	54.5	47.2	50.5

The best strategy to accelerate the computation of the sensitivity matrix is to use in-plane symmetries, to disregard atomic operations, and to reorder the attenuation coefficients and sensitivity matrices in order to make the in-plane symmetrical voxels neighbours as explained in section 7.4.3. Table 7.I shows also that in the case where accesses are coalesced, using atomic floating point operations add a 6 to 8-fold penalty on the computation time.

Ignoring atomic operations might lead to race conditions and affect the accuracy of the resulting image. To assess the effect of using or not atomic operations for this particular problem, images obtained with and without atomic operations were compared. Figure 7.4 shows the relative difference image, where the maximal value is 0.8%.

Consequently, is seems that atomic operations are not absolutely necessary for PET recon-



a) b)

Figure 7.4 – Relative difference images of a homogeneous phantom, computed with and without atomic operations. a) Central slice and $2 \times 2 \times 3.15$ mm^3 voxels and b) border slice and $4 \times 4 \times 3.15$ mm^3 voxels.

struction given their high computational cost. Even if race conditions can happen, their occurrence is relatively rare for this particular problem.

Table 7.II presents the execution times of sensitivity matrix calculations on a 2.40 GHz Intel Core2 Quad Q6600 CPU (1 core used) using STIR along with GPU results using coalesced accesses and no atomic operations. GPU/CPU acceleration factors are also given. Although this comparison is not the same operation-wise, it gives an idea of the speedup achievable with GPUs for this class of problems.

Tableau 7.II – Execution times in seconds for sensitivity matrices calculations on CPU and GPU using in-plane symmetries, disregarding atomic operation and coalescing memory accesses.

	$188 \times 188 \times 57$ matrix of $2 \times 2 \times 3.15$ mm^3			$144 \times 144 \times 57$ matrix of $4 \times 4 \times 3.15$ mm^3		
Tangential sub-LORs	1	2	3	1	2	3
CPU Execution time (s)	631.2	842.9	903.1	687.5	816.2	862.6
GPU Execution time (s)	2.8	5.6	8.3	2.5	4.8	7.2
GPU/CPU acceleration factor	225	150	112	275	170	119

7.5.2 Computation of LM-OSEM algorithm

Table 7.III reports the computation times of the LM-OSEM implementation on GPU and on CPU. The results reported in this table are similar to those obtained by Pratx et al. [172] for a

similar problem. The computing time per event of the LM-OSEM algorithm is longer than the computing time per LOR for the senstivity matrix for which in-plane geometrical symmetries are used to accelerate the computation. Since the list-mode data are not arranged in any regular pattern, the geometrical symmetries can be used to accelerate the reconstruction but the access to the estimated and gradient images during the reconstruction is a random access that cannot be optimized.

Tableau 7.III – Computation time in seconds of the LM-OSEM algorithm on GPU and CPU for one million of events and one iteration.

	$188 \times 188 \times 57$ matrix of $2 \times 2 \times 3.15$ mm^3			$144 \times 144 \times 57$ matrix of $4 \times 4 \times 3.15$ mm^3		
Tangential sub-LORs	1	2	3	1	2	3
Axial sub-LORs	2			2		
GPU	0.35	0.73	1.1	0.39	0.76	1.2
CPU	32.7	36.0	38.9	39.6	47.9	52.8
Acceleration	93	49	35	101	63	44

To further explore the effect of ignoring atomic operations to compute the sensitivity matrix, the contrast phantom described in section 7.4.5.3 was reconstructed with and without this feature. The results, reported in figure 7.5, suggest that errors arising from race conditions in the calculation of the sensitivity matrix lead to small differences in the final image.

This observation is also supported by figure 7.7 that presents the CRC versus the number of tangential sub-LORS in a reconstruction with $2 \times 2 \times 3.15$ mm^3 voxels. This figure shows that the CRC change is negligible for tangential sub-LORs numbers beyond 3 and supports our hypothesis regarding the optimal number of sub-LORs per detector pair.

7.6 Discussion and conclusion

The senstivity matrix that accounts for attenuation and normalisation factors was computed on a Tesla C2050 GPU in less than 10 seconds for a Philips Gemini PET scanner using four to six times the intrinsic number of LORs for this system, which is approximately 85 millions. To efficiently compute the sensitivity matrix on the GPU, the geometrical symmetries of the scanner were exploited. Furthermore, each thread was associated to one basis LOR, and computed both the forward and back-projection of its symmetrical LORs. Therefore, the computation performance was improved by cutting down the access to global memory to store the forward projection values for all the LORs. This strategy allows also to free more memory space to load the event

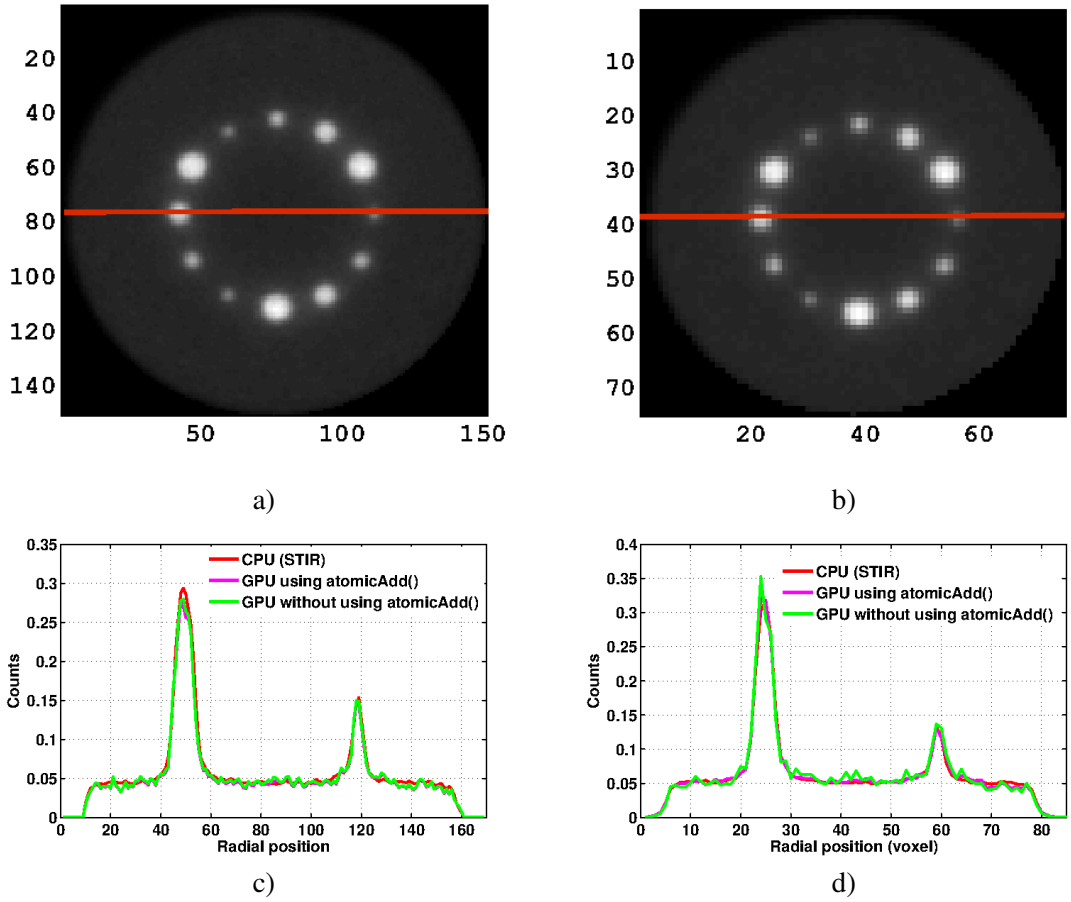
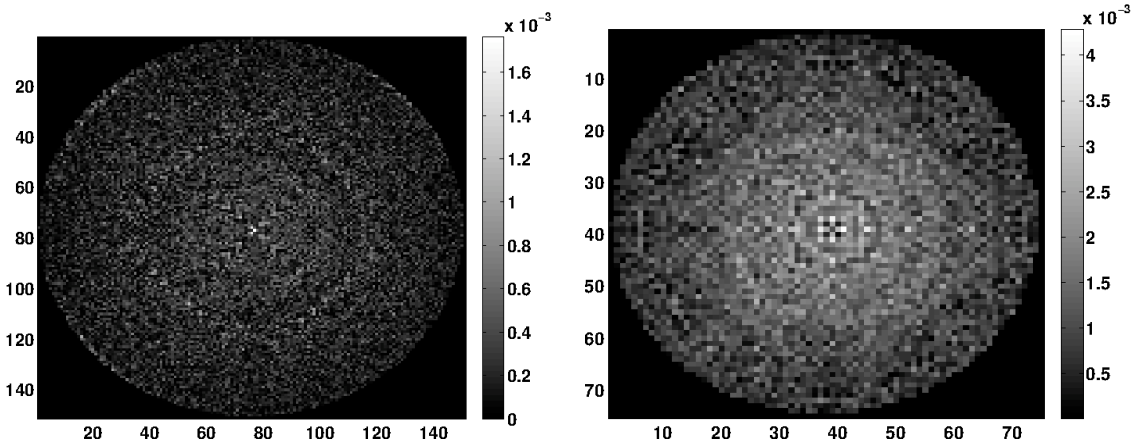


Figure 7.5 – Transverse slices computed by the LM-OSEM algorithm using one iteration, 10 frames and 113 millions of events : a) central slice using atomic operations and $2 \times 2 \times 3.15$ mm^3 voxels , b) 4.5 cm off-center slice without using atomic operations and $4 \times 4 \times 3.15$ mm^3 voxels, c) and d) corresponding cut-views.



a) b)

Figure 7.6 – Relative difference images showing the effect of atomic operations : a) central slice and $2 \times 2 \times 3.15$ mm^3 voxels, b) 4.5 cm off axial slice and $4 \times 4 \times 3.15$ mm^3 voxels.

stream from the CPU memory to the GPU global memory as the GPU computes the sensitivity matrix.

The normalisation data matrix was stored in the global device memory as linear memory and reordered to allow coalesced GPU accesses. Similarly, the 3D attenuation coefficient matrix was stored as a 3D CUDA array matrix and associated to a 3D texture and was reordered so that the in-plane symmetrical voxels become neighbours. The sensitivity matrix was stored as linear memory and mapped in the same manner as the attenuation coefficient matrix.

It has been shown that without reordering the two matrices, using the in-plane symmetries does not improve the efficiency.

Shared memory was used to store the forward projection values for the active threads and the requirements for coalescence were respected.

The use of atomic operations led to a sixfold time penalty for the case where symmetries were used and the matrices were reordered for coalesced accesses. Since the relative difference between images reconstructed using senstivity matrix computed with and without atomic operations was less than 0.8 % (figure 7.4 and 9.5), this feature seems not necessary to compute senstivity matrix for LM-OSEM algorithm. However, more comparisons using clinical data are necessary to make this conclusion more robust. A fully 3D LM-OSEM algorithm was implemented on a Tesla C2050 GPU device, including the calculation of the sensitivity matrix. The reported reconstruction times for a Gemini GXL PET are not only compatible with a clinical use

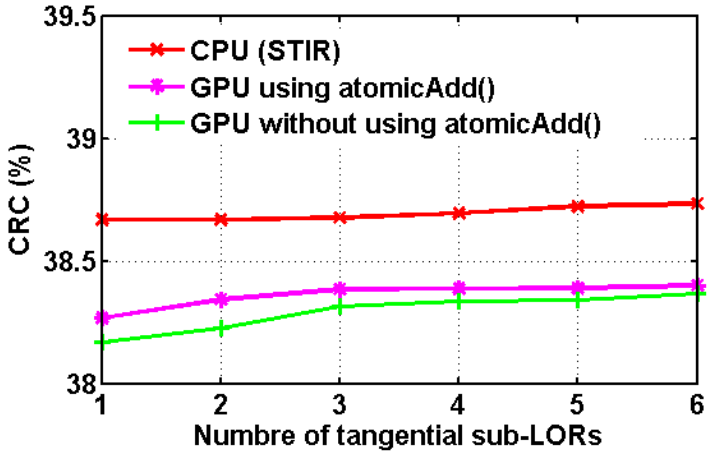


Figure 7.7 – The CRC versus the number of tangential sub-LORS calculated for the 17 mm hot lesion in the central slice of the phantom described in section 7.4.5.3, computed using one iteration, 10 frames, 113 millions of events and $2 \times 2 \times 3.15$ mm^3 voxels.

of 3D LM-OSEM algorithms, but also allow advanced list-mode PET reconstructions for routine clinical work. Possible applications include real time list-mode gated reconstructions and 4D list-mode reconstructions. Further work is required to account for random and scatter events, which are time-consuming in list-mode reconstruction.

One solution to further accelerate the LM-OSEM algorithm is to use a multiscale grid and an adaptative multiple rays per detector pairs approach [133]. This consists in starting the reconstruction during the acquisition using first a coarse sampling of the object with a single ray per detector pair, and then to refine the grid as the statistics improve and to adapt the number of rays used per LOR with the new grid.

Although list-mode was discussed here, frame-mode reconstruction might potentially be more efficient for relatively long frames when appropriate methodology is employed [77]. This avenue will also be explored, in the context of our previous work on GPU-accelerated histogram-mode PET reconstruction [142, 143]. Histogram-mode allows to use symmetries and to coalesce the access to data, which might lead to further acceleration. Furthermore, for 4D reconstructions, the concept of timogram introduced by Nichols et al. [149] to compress list-mode data might be well suited for a GPU implementation, again by allowing a coalesced access to data. This kind of approach will be explored in the future.

Acknowledgements

This work was supported by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT) and by the Natural Sciences and Engineering Research Council of Canada (NSERC). NVIDIA Corporation kindly donated material to conduct this study.

CHAPITRE 8

FAST GPU-BASED COMPUTATION OF SPATIAL MULTIGRID MULTIFRAME LMEM FOR PET

Accepté pour publication dans le journal Medical & Biological Engineering & Computing (soumis le 3 avril 2014, accepté avec correction le 24 juillet 2014, révisé et resoumis le 25 août 2014, et accepté pour publication le 22 mars 2015.)

Auteurs : Moulay Ali Nassiri[1], Jean-François Carrier[1] et Philippe Després[2].

[1]*Département de radio-oncologie, Centre hospitalier de l'Université de Montréal (CHUM), Montréal (Québec), CANADA.*

[2]*Département de radio-oncologie, Centre hospitalier universitaire de Québec (CHUQ), Québec (Québec), CANADA et Département de physique, de génie physique et d'optique, Université Laval, Québec, CANADA.*

Contribution des auteurs :

Moulay Ali Nassiri est le principal auteur de cet article. Il a apporté la plus grande contribution au projet, spécifiquement :

- la définition de la problématique et des objectifs de ce travail ;
- le développement et la parallélisation de l'algorithme Multigrid Multiframe list-mode expectation maximisation (MGMF-LMEM) ;
- l'invention du critère de convergence utilisant le gradient de la fonction objective
- l'implémentation et l'optimisation sur GPU l'algorithme MGMF-LMEM ;
- l'analyse des résultats et la rédaction du manuscrit publié.

Jean-François Carrier et Philippe Després ont été les encadrants principaux de ce travail. Ils ont aussi restructuré, corrigé le manuscrit initial et final et ils ont participé à formuler les réponses aux commentaires des arbitres.

8.1 Résumé et mise en contexte

– **Problématique et objectifs** : Malgré ses avantages par rapport à la reconstruction en mode sinogramme, le mode liste n'est pas encore utilisé en clinique de routine, comme nous l'avons déjà signalé dans le chapitre 7, à cause du temps de reconstruction qui est très long par rapport aux besoins cliniques. Ainsi, plusieurs travaux ont été effectués cette dernière décennie pour accélérer la reconstruction à partir de ce mode sur GPU [45, 57, 170–172, 198, 251]. Cependant, d'une part, ces travaux ont porté essentiellement sur l'algorithme LM-OSEM qui est la version accélérée mais non convergente de l'algorithme convergent de référence LM-EM [30, 82, 120, 179, 184, 231]. D'autre part, ces travaux ont ignoré le fait que le temps de calcul de la matrice de sensibilité est long aussi et qu'il constitue alors un obstacle à l'introduction du mode liste en clinique.

  Ainsi, pour participer à cet effort de la communauté scientifique d'accélérer le temps de reconstruction à partir du mode liste, nous avons au niveau de notre travail antérieur présenté au chapitre 7 [145], mis l'accent sur l'accélération sur GPU de la matrice de sensibilité. Le temps de calcul de cette matrice obtenu est de moins de 10 secondes pour le tomographe Philips Gemini GXL de 85 millions LORs. Pars ailleurs, nous avons aussi noté que le temps d'exécution par événement et par itération des algorithmes mode liste est au moins 8 fois supérieur au temps d'exécution par LOR et par itération des algorithmes de reconstruction à partir du mode sinogramme. Ceci s'explique par le fait que contrairement à la reconstruction à partir des sinogrammes, la reconstruction mode liste ne permet ni d'utiliser les symétries géométriques et ni de respecter les conditions de coalescence de la mémoire globale. Donc, comme continuation à notre travail présenté au chapitre 7, nous avons développé dans ce travail la nouvelle stratégie MGMF-LMEM qui permet d'esquiver les obstacles d'accélération sur GPU de la reconstruction mode liste. L'objectif est d'accélérer l'algorithme LM-EM à un niveau qui permet de faire de la reconstruction dynamique en temps réel. Le choix de LM-EM est justifié par le fait que c'est un algorithme de référence qui est convergent et robuste et qui a été délaissé du fait que sa convergence est très lente en comparaison à l'algorithme LM-OSEM.

– **Méthodologie** : l'approche MGMF-LMEM développée est une extension au mode liste de l'algorithme mode sinogramme *multigrid expectation-maximisation* (MGEM) proposé par Ranganath *et al.* [187]. L'algorithme MGMF-LMEM implémenté consiste à diviser,

par exemple, l'intervalle d'acquisition $[T_0, T_{aq}]$ en 3 phases $[T_0, T_1]$, $[T_1, T_2]$ et $[T_2, T_{aq}]$ et ensuite à : 1) commencer à l'instant T_1 l'exécution de l'algorithme LM-EM en utilisant les événements de coïncidences collectés durant l'intervalle $[T_0, T_1]$. La reconstruction se fait sur une matrice de faible définition (grands voxels) $Grid_0$ car les statistiques sont faibles à cette phase, 2) attendre jusqu'à ce que le critère de convergence de LM-EM soit validé et interpoler l'image reconstruite afin d'obtenir cette dernière sur une nouvelle grille $Grid_1$ présentant une meilleure définition (petits voxels), 3) attendre jusqu'à l'instant T_2, puis exécuter de nouveau l'algorithme LM-EM durant la phase 3 pour les événements collectés durant l'intervalle $[T_0, T_2]$ et en utilisant comme matrice d'initialisation l'image obtenue par interpolation de la matrice reconstruite durant la phase précédente, et 4) reprendre le même processus en réinitialisant l'exécution de l'algorithme LMEM lorsque l'acquisition est terminée à T_{aq}. La reconstruction durant cette dernière phase sera faite sur la grille finale plus fine $Grid_3$ en utilisant tous les événements d'acquisition et l'image d'initialisation obtenue par interpolation de l'image construite durant la phase précédente.

Le plus grand défi de l'algorithme MGMF-LMEM est d'assurer de la convergence de LM-EM exécuté durant une phase avant sa fin. De ce fait, nous avons : i) utilisé la technique multi-trajectoires adaptive pour calculer la MS et qui consiste à utiliser une seule trajectoire par LOR pour la matrice de faible définition $Grid_0$, et à multiplier par 2 le nombre de trajectoires chaque fois qu'on change de définition pour passer des grands vers des petits voxels ; ii) introduit un critère de convergence dont le temps de calcul par GPU est négligeable ; iii) implémenté des méthodes d'interpolation bien adaptées au calcul sur GPU, telles que la méthode du voisin le plus proche, l'interpolation bilinéaire et la méthode de l'interpolation cubique B-spline déterminée à partir de la méthode bilinéaire, et iv) utiliser pour MGMF-LMEM les mêmes stratégies d'implémentation sur GPU développées dans le chapitre précédent (paragraphe 7.4.4) pour l'algorithme LM-OSEM.

– **Résultats et discussion** : les résultats obtenus montrent que l'algorithme MGMF-LMEM converge vers la même solution que LM-EM avec une vitesse qui est 3 fois plus rapide que ce dernier. Le temps d'exécution sur la carte GPU Tesla 2050C des algorithmes MGMF-LMEM et LM-EM est 1.1 secondes par millions d'événements de coïncidences et par itération pour une grille de définition de 188x188x 57 voxels de taille 2x2x3.15 mm^3. Cette vitesse de calcul permet d'exécuter une itération de MGMF-LMEM pour 60 millions d'événements en 66 secondes. Ces résultats montrent aussi que pour des lésions chaudes

de taille 17 et 22 mm, l'algorithme MGMF-LMEM permet d'obtenir durant une itération un coefficient de recouvrement de contraste (*contrast recovery coefficients*) supérieur à 75% du recouvrement de contraste maximal, et d'avoir la plus petite erreur moyenne au sens des moindres carrés entre l'image estimée et l'image réelle. Par conséquent, l'algorithme MGMF-LMEM peut être utilisé comme un algorithme d'un seul passage pour faire des reconstructions très rapides pour les longues acquisitions et des reconstructions en temps quasi réel pour les acquisitions dynamiques. Cependant, malgré que les résultats obtenus soient prometteurs, une évaluation en milieu clinique est nécessaire pour optimiser le nombre de phases, de grilles et la valeur du critère de convergence. Cette évaluation doit être effectuée après l'implémentation des fonctions de correction des événements diffusés et fortuits. Pour mettre en oeuvre ces fonctions de corrections sans détériorer les performances de calcul de l'algorithme MGMF-LMEM, il serait très judicieux d'utiliser une deuxième carte GPU dédiée à ces algorithmes de correction comme proposée par Wang [234] et d'estimer le diffusé en utilisant le logiciel Monte Carlo sur GPU développé par notre groupe [75].

8.2 abstract

Significant efforts were invested during the last decade to accelerate PET list-mode reconstructions, notably with GPU devices. However, the computation time per event is still relatively long and the list-mode efficiency on the GPU is well below the histogram-mode efficiency. Since list-mode data are not arranged in any regular pattern, costly accesses to the GPU global memory can hardly be optimized and geometrical symmetries cannot be used.To overcome obstacles that limit the acceleration of reconstruction from list-mode on the GPU, a multigrid and multiframe approach of an expectation-maximization algorithm was developed. The reconstruction process is started during data acquisition and calculations are executed concurrently on the GPU and the CPU while the system matrix is computed on-the-fly.A new convergence criterion also was introduced, which is computationally more efficient on the GPU. The implementation was tested on a Tesla C2050 GPU device for a Gemini GXL PET system geometry. The results show that the proposed algorithm (multigrid and multiframe list-mode expectation maximisation, MGMF-LMEM) converges to the same solution as the list-mode expectation-maximization (LMEM) algorithm more than three times faster. The execution time of the MGMF-LMEM al-

gorithm was 1.1 s per million of events on the Tesla C2050 hardware used, for a reconstructed space of $188 \times 188 \times 57$ voxels of $2 \times 2 \times 3.15$ mm^3. For 17 and 22 mm simulated hot lesions, the MGMF-LMEM algorithm led on the first iteration to contrast recovery coefficients (CRC) of more than 75 % of the maximum CRC while achieving a minimum in the relative mean square error. Therefore, the MGMF-LMEM algorithm can be used as a one-pass method to perform real-time reconstructions for low-count acquisitions, as in list-mode gated studies. The computation time for one iteration and 60 millions of events was approximately 66 s.

8.3 Introduction

Reconstruction algorithms used for Positron Emission Tomography (PET) in the clinic are typically based on histogrammed data. This acquisition mode is favored over list-mode for most clinical applications, where the number of recorded events is significantly larger than the number of line-of-responses (LORs). Histogrammed data are often compressed to reduce the sinogram size and to accelerate the image reconstruction [180, 226]. This acquisition mode also allows the use of geometrical symmetries for the computation of the system matrix (SM) that globally accelerate the reconstruction process [77, 80, 88, 145, 195, 256].

Various strategies were used to accelerate the reconstruction from histogrammed data. One consists in using several spatial resolution grids to recover distinct frequency ranges of the estimated image instead of using a single fine grid that leads to lengthy calculations. This multigrid expectation-maximisation (MGEM) algorithm was proposed by Ranganath *et al.* in 1983 for histogram-based reconstructions [187]. Raheja *et al.* have extended this approach and developed a multiresolution algorithm in projection space [182]. It was shown that these approaches provide faster convergence than a single grid maximum-likelihood expectation-maximisation algorithm (MLEM). Furthermore, the work of Ho *et al.* shows that a multigrid inversion approach results in significant improvement in convergence speed compared to the fixed-grid for Bayesian reconstruction algorithms in transmission and emission tomography [154, 155]. More recently, Mendes *et al.* have also used the concept of multigrid for histogram-mode reconstructions [134]. This work extends the MGEM algorithm to the multiframe approach in which the reconstruction is started during the acquisition using a coarse sampling of the object that is refined as the number of counts increases. Mendes *et al.* have shown that this approach accelerates the convergence compared to a single grid MLEM algorithm and that it can be used to perform near real-time

reconstructions.

The reconstruction from list-mode data has gained interest during the last decade. This mode offers many advantages over the histogram mode such as convenience and accuracy of motion correction, the straightforward use of time-of flight (TOF) information to improve the reconstruction accuracy, the possibility of using temporal basis functions to improve 4D image reconstructions, and overall better temporal resolution, contrast and noise properties [183]. However, the reconstruction from list-mode is relatively long especially for large acquisitions in which the number of events is larger than the number of LORs of the system. The long time required to process the sensitivity matrix for list-mode reconstructions is also an obstacle to the introduction of this approach in the clinic. In fact, the sensitivity matrix must be calculated for each patient based on its own attenuation map obtained from a transmission scan [36, 66, 125, 176, 184].

Significant efforts were recently devoted to the acceleration of list-mode data reconstruction on Graphics Processing Units (GPUs) [13, 14, 24, 45, 57, 170–172, 198, 251]. Nevertheless, the computation time per event is still relatively long and forbids real-time reconstructions from list-mode. Previous attempts at accelerating list-mode reconstructions were mostly focused on ordered-subset expectation-maximization (LM-OSEM) algorithms, which do not converge in general [30, 82, 120, 179, 184, 231] and in consequence might hinder quantification efforts. In this context, a fast and converging list-mode reconstruction algorithm is highly desirable.

In our previous work, the GPU device was used to compute the sensitivity matrix for a list-mode algorithm in less than 10 seconds for a PET scanner with approximately 85 million LORs (Philips Gemini GXL) [145]. Still, the processing time per event for the list-mode reconstruction was more than eight times longer than the time required per LOR for the histogram-based reconstruction [145]. Since list-mode data are not arranged in any regular pattern, the geometrical symmetries cannot be used to accelerate the reconstruction and the access to the matrices stored in the GPU global memory during the reconstruction is a random process that can hardly be optimized [54, 174].

In this work, we have implemented on GPU a multigrid/multiframe approach of an expectation-maximization algorithm for list-mode acquisitions (MGMF-LMEM), following previous work on this subject by Mendes *et al.* [135]. The objective is to develop new strategies to accelerate the reconstruction from list-mode, at a level that allows near real-time execution. The choice of the LM-MLEM algorithm over a LM-OSEM algorithm is motivated by the convergent and robust nature of the LMEM algorithm.

The proposed MGMF-LMEM algorithm processes events on-the-fly during the acquisition with a coarse sampling of the object and one ray per detector pair to compute the SM. As the number of counts increases, the grid is refined and the number of rays used per LOR is increased. To interpolate the estimated image from a low resolution grid to a finer one, different methods were used : nearest-neighbor, linear and cubic B-spline interpolations [110, 194]. The effects of these methods on the quality of the reconstruction and on the computation time were evaluated. A Gaussian filter was also implemented on the GPU to process the image before re-sampling the object.

The evaluation of the MGMF-LMEM algorithm was done using simulation data modeling a Philips Gemini GXL PET scanner. Corrections for random, scatter and time-of flight events were not implemented at this time.

8.4 Theory and Methods

8.4.1 List-mode expectation-maximization

Considering the linear relationship between the activity distribution of radiotracer in the object and the detection system, and that the object is parametrized by a set of voxels $V = \{1,2,...,J\}$, we have

$$d_i = \sum_{j=1}^{J} p_{i,j}\lambda_j + n \tag{8.1}$$

where λ_j is the number of events emitted from voxel j, d_i is the number of coincidences detected in LOR i and n is the statistical noise, and the coefficients $p_{i,j}$ are the elements of the system matrix (SM) $\mathbf{P}$ defined as the detection probability for a photon pair created in voxel j along the sinogram bin i ($i = 1..I$). In this work, SM is computed on-the-fly with the Siddon raytracing algorithm [209] using multiple rays per detector pair [145].

The list-mode expectation-maximization (LMEM) algorithm estimates the activity of radiotracer in the patient λ_j from the individual list-mode data events. LMEM is derived from the MLEM algorithm by replacing the sum over LORs with a summation over events [20, 190], and is given by

$$\lambda_j^m = \frac{\lambda_j^{m-1}}{S_j} \sum_{k=1}^{M} p_{i_k,j} \frac{1}{\sum_{j=1}^{J} p_{i_k,j} \lambda_j^{m-1}} \tag{8.2}$$

$$S_j = \sum_{i=1}^{I} p_{i,j} a_i \varepsilon_i \tag{8.3}$$

where M is the number of measured events, i_k is the LOR related to the k^{th} list-mode event, S_j is the element corresponding to voxel j of the sensitivity matrix $\mathbf{S}$ that accounts for variations due to detector efficiency and attenuation in the patient, a_i are the attenuation correction factors calculated by forward-projecting along all LORs the patient attenuation coefficient map obtained from the transmission scan and ε_i is the normalization factor. S_j is computed using all possible LORs passing through the explored region [145].

The list-mode maximum-likelihood L_{List} [85] is

$$L_{List} = \ln(Pr(\mathbf{d}|\lambda)) = \sum_{k=1}^{M} \ln(\mathbf{P}\lambda)_k - \sum_{j=1}^{J} (S_j \lambda_j). \tag{8.4}$$

The gradient of L_{List} is given by

$$\frac{\partial L_{List}}{\partial \lambda_j} = G_j - S_j$$

$$G_j = \sum_{k=1}^{M} p_{i_k,j} \frac{1}{\sum_{j=1}^{J} p_{i_k,j} \lambda_j} \tag{8.5}$$

The matrix $\mathbf{G}$ converges to the sensitivity matrix $\mathbf{S}$ [20, 61] because the LMEM algorithm is a convergent algorithm.

8.4.2 Multigrid and multiframe list-mode expectation-maximization

The MGMF-LMEM is the extension to list-mode of the multiscale/multiframe reconstruction algorithm from histogram data proposed by Mendes *et al.* [134]. The MGMF-LMEM requires a set of pyramidal grids $[Grid_n, Grid_{n-1}, .., Grid_0]$ sampling the same object with different resolutions. The voxel sizes of the $Grid_{i-1}$ in the axial plane is the same or larger than in the $Grid_i$ grid. The sampling in the axial direction (Z) is kept unchanged for all grids.

A predefined set of discrete time points $[T_0, T_1, .., T_n]$ over the length T_n of the acquisition is also required. The reconstruction starts during the acquisition at the time T_0 with the LMEM

algorithm and the coarse image definition $Grid_0$, using all events acquired up to this time. When the convergence criterion is satisfied, the output image is interpolated to the next grid definition $Grid_1$ and used as the initial image for the next round of the LMEM algorithm that uses events occurring up to the time T_1. The processes is repeated until the end of the acquisition according to

$$\lambda_j^{m,T_s} = \frac{\lambda_j^{m-1,T_s}}{S_{j,Grid_s}} \sum_{k=1}^{M_{T_s}} a_{i_k,j} \frac{1}{\sum_{j=1}^{J} a_{i_k,j} \lambda_j^m} \tag{8.6}$$

$$\lambda_j^{0,T_s} = W_s(\lambda_j^{k,T_{s-1}}) \tag{8.7}$$

where λ_j^{m,T_s} is the intensity value in voxel j estimated after m iterations using the M_{Ts} events acquired until T_s for the corresponding image definition $Grid_s$. $\lambda_j^{k,T_{s-1}}$ is the output image estimated during the previous computation of LMEM ($Grid_{s-1}$) using k iterations. The interpolation operator W_s is applied to this image to estimate the initial image λ_j^{0,T_s} for the current grid. The $S_{j,Grid_s}$ is the element j of the sensitivity matrix corresponding to image definition $Grid_s$. All sensitivity matrices corresponding to different grids are computed after the transmission scan and before starting the first reconstruction at T_0.

The implementation of MGMF-LMEM requires a stopping criterion during each phase. Many different stopping criterion for the MLEM algorithm have been proposed [25, 61, 121], some of them based on the properties of the updating coefficients

$$C_j^m = \frac{1}{S_j} \sum_{k=1}^{M} p_{i_k,j} \frac{1}{\sum_{j=1}^{J} p_{i_k,j} \lambda_j^{m-1}} \tag{8.8}$$

which converges to 1 [61]. In this work, a new stopping criterion (*SC*) based on the gradient matrix **G** is proposed. In fact, the matrix **G** (equation 2) converges to the sensitivity matrix **S** [20, 61]. Therefore, since the **G** matrix is computed during the reconstruction at each iteration in order to update the image λ as shown in equation 2, it is sensible to use this matrix to determine the stopping criterion *SC*. The Frobenius norm of **G** was used to define the *SC* as

$$SC = \frac{||\mathbf{G}^{m+1}||}{||\mathbf{G}^m||} \tag{8.9}$$

where $\mathbf{G^m}$ is the matrix **G** computed at iteration m and its Frobenius norm $||\mathbf{G}|| = \sqrt{\sum_{j=1}^{J} G_j^2}$

is obtained using texture access on the GPU to accelerate the process. The choice of this stopping criterion is highly motivated by a fast and efficient GPU implementation.

8.4.3 Interpolation

The resampling of the reconstructed image from the current grid to the next finer grid is an interpolation problem. Using high-order and large sizes for interpolation kernels potentially leads to more accurate results. In return, the complexity of implementation and the associated computing time increases with the order and the size of kernels. Three interpolation methods were tested here, based on the previous work of Lehmann *et al.* [110] : nearest-neighbor, bilinear, and cubic B-spline.

The nearest-neighbor interpolation method is the simplest but typically leads to strong aliasing and blurring effects.

The bilinear interpolation is a modest low-pass filter that offers better frequency characteristics than the nearest-neighbor method, but that attenuates high-frequencies below the cut-off point potentially resulting in excessive smoothing and aliasing in the interpolated image [161]. This method is often used in medical imaging due to its ease of implementation and its fast computation. Bilinear interpolation of images is well-suited for GPUs as this operation is hardware-implemented in these devices.

The cubic B-spline method has better characteristics than bilinear interpolation and than high-order polynomial interpolation. It is a reasonably good low-pass filter that has a good stopband response but that can smooth more than necessary in the passband. Consequently, cubic B-spline improves the quality of the interpolation but increases the smoothing effects [110]. Cubic B-spline interpolation can be efficiently implemented on the GPU as a combination of several hard-wired bilinear interpolations [194] and this approach was used in this work.

For non-regularized iterative stochastic algorithms such as LMEM, the noise increases with the number of iterations [39, 68, 95, 100, 183, 215, 242]. In the current implementation, the estimated image was smoothed by a 3D Gaussian filter for a sigma of 4 mm before interpolation in order to mitigate the effect of noise.

8.4.4 System matrix computation

In the MGMF-LMEM algorithm, the system matrix was computed on-the-fly on the GPU with a raytracing approach using mutliple rays per detector pair (sub-LORs) [145]. Two axial sub-LORs were used while the number of tangential sub-LORs depended on the tangential size of the image definition $Grid_s$ involved. One, two and three tangential sub-LORs were used respectively for grids having voxel sizes larger than the tangential size, larger than half the tangential size and smaller or equal to half the tangential size of detectors. This strategy led to an optimal sampling of the image based on the trade-off between image quality and computation time [144].

8.5 GPU implementation

In this work, a NVIDIA Tesla C2050 GPU was used. This device has 448 cores and 3 GB of VRAM at 144 GB/s of bandwidth. Since it supports CUDA compute capability 2.0, the global memory is cached up to 48 kB per multiprocessor and supports the floating-point atomic addition operating on 32-bit words in global and shared memory. This last feature is very important in PET image reconstruction since it allows the reduction of data loss in the calculation of the back projection due to simultaneous read-modify-write operations associated with parallel threads. For compilation, `gcc` version 4.4.1 for the C++ code and `nvcc` version 3.2 for the GPU code were used.

8.5.1 Implementation of the MGMF-LMEM algorithm

Six execution kernels were used to implement the MGMF-LMEM algorithm on the GPU. The first one computes the sensitivity matrices that account for normalization and attenuation in the patient for the specific grids used. The implementation on the GPU of this kernel was the principal objective of our previous study [145]. The second kernel calculates the matrix $\mathbf{G^m}$ as defined in equation 2 at each iteration while the third one simply updates the current image $\boldsymbol{\lambda}^{\mathbf{m}}$ through a voxel by voxel multiplication by the image $\mathbf{G^m}$ followed by a normalization using the sensitivity matrix. This strategy, used to compute the LMEM algorithm efficiently on the GPU, was detailed in a previous paper [145].

The fourth execution kernel was dedicated to the interpolation from the current to the finer grid. The implementation on the GPU of cubic B-spline interpolation was an extension in 3D of the work presented by Ruijters *et al.* [194]. The image was stored in a 3D texture and the cubic

B-spline was decomposed into eight linear interpolations that were performed in hardware on the GPU. The nearest-neighbor and linear versions were implemented with native CUDA functions.

The fifth kernel was used to Gaussian-blur the image before interpolation. The implementation of this kernel was done in the frequency domain using the CuFFT library from NVIDIA.

Finally, the sixth kernel computes the Frobenius norm of the matrix $\mathbf{G^m}$, used as a convergence criterion.

The reconstruction begins by launching the sensitivity kernel n times where n is the number of grids used. Since the kernel launches are asynchronous, the control is returned to the CPU host before the computation of the sensitivity matrices is completed. This permits to the CPU to reading simultaneously the event stream of the first frame from the hard disk and to transfer this data to the global GPU memory. Then, the kernels computing the gradient and updating the image are launched as long as the convergence criterion is not satisfied. When this criterion is reached, the estimated image is filtered and interpolated to a finer grid with the corresponding execution kernels. The cycle then repeats with the next data frame, which is loaded in the GPU memory while filtering/interpolation operations are completing. This strategy allows to mask the long transfer time from the CPU memory to the GPU memory, which is a bottleneck in this application.

8.5.2 Grids and frames used

Figure 8.1 shows schematically the acquisition and reconstruction sequence for the MGMF-LMEM algorithm. The reconstruction was performed for a final image size ($Grid_1$) of $188 \times 188 \times 57$ voxels of $2 \times 2 \times 3.15$ mm^3. The grid sequence used was ($Grid_0, Grid_1, Grid_1$) for three frames delimited by $[0, T/4, T/2, T]$ over the total acquisition time T. The coarse $Grid_0$ grid was made of $94 \times 94 \times 57$ voxels of $4 \times 4 \times 3.15$ mm^3. The first phase of reconstruction (Phase 1) was started at time $T/4$ for $Grid_0$ using all events acquired during the time interval $[0, T/4]$ and a uniform distribution as the initial image. The computation stopped when the convergence criterion was satisfied. The second phase of computation (Phase 2) started at $T/2$ for $Grid_1$ using all events acquired during the time interval $[0, T/2]$ and using as initial image the reconstructed image obtained during the previous phase after it was interpolated on the grid $Grid_1$. The last phase of reconstruction (Phase 3) started at T for grid $Grid_1$ using all events acquired during the time interval $[0, T]$ and using the reconstructed image during the previous phase as the initial image.

To compute the SM, two tangential sub-LORs and two axial sub-LORs were used for $Grid_0$ while three tangential sub-LORs and two axial sub-LORs were used for $Grid_1$. These choices were shown to present the best trade-off between the error on the quantification and the computing time [144].

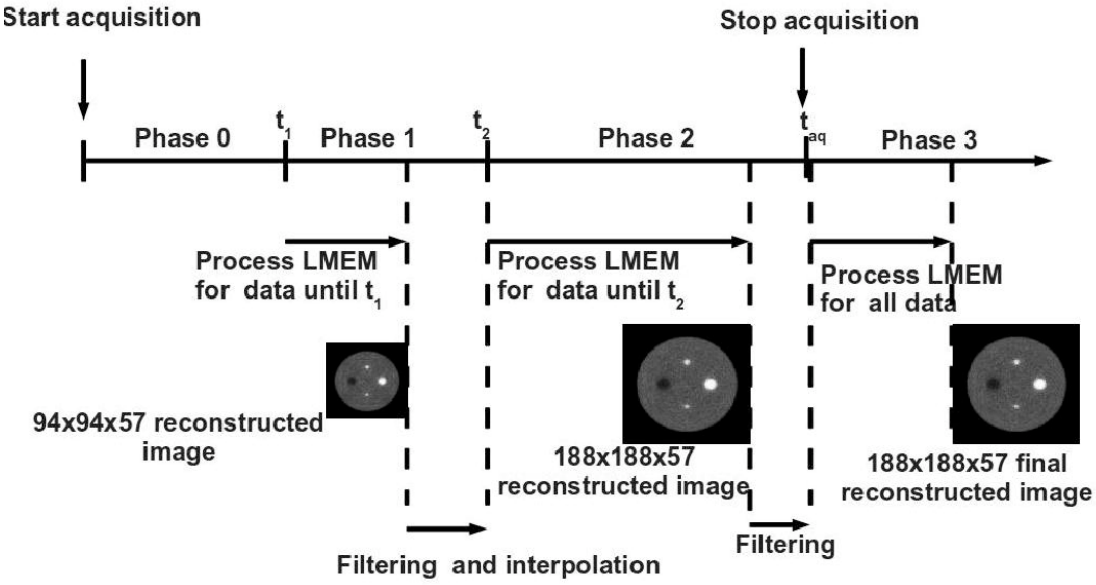


Figure 8.1 – Schematic diagram of the MGMF-LMEM algorithm used in this work.

8.6 Monte Carlo simulations and quantitative measurements

8.6.1 Gemini GXL PET

Monte Carlo simulations of list-mode data acquisitions were performed with the GATE package [90] for the Philips Gemini GXL PET system based on the work of Lamare et al. [106]. This scanner is composed of 17,864 individual GSO crystals of 4, 6 and 20 mm sizes in tangential, axial and radial directions respectively. These crystals are organized in 28 blocks, each being a set of 22 tangential and 29 axial individual GSO crystals. This scanner offers a field of view of 56 cm radially and 18 cm axially. In the simulations, coincidence is allowed between each of the 28 blocks and the opposing 15 blocks for an axial maximum ring difference of 28 detectors. To generate the prompt events, coincidences are defined with an energy window of 410-665 keV and a time window of 7.5 ns. A non-paralysable dead time of 80 ns was applied to the singles

output after energy cut to model the decay time of GSO and the integration time of all the scintillation light. After generating coincidences, another non-paralysable dead time of 80 ns was applied to model the dead time of the digital coincidence circuit. For each event, the identity of the two detectors defining the LOR was stored in list-mode format (LMF) [130] along with energy and timing information. The simulation was performed on a cluster of 128 Intel Xeon E5472 quadcore processors clocked at 3 GHz. For each realisation, 32 independent simulations were launched according to the approach proposed by De Beenhouwer et al. [48]. This strategy allows to reproduce physical properties such as singles rates, random coincidence events and system dead time, and allows to optimise the speed of the computation.

8.6.2 Phantom

The LMEM and the MGMF-LMEM algorithms were tested on simulated data of a cylindrical phantom having spherical lesions of different diameters and activities. 65 millions of true events were simulated : 27 millions during phase 0, 20 millions during phase 1 and 18 millions during phase 2.

The phantom has a diameter of 30 cm, a length of 20 cm, and is filled with 2.2 mCi of ^{18}F. The lesions are inserted on the central axial plane and on axial planes located at 4.5 cm off-center on each side of the central plane. The central axial plane contains one spherical, 50 mm diameter cold lesion at its center and 12 spherical hot lesions of sizes 10, 13, 17, and 22 mm of contrast 4 : 1 positioned at 7 cm radially from the center, in a pattern that is repeated three times. The planes located 4.5 cm off-center contain 3 spherical hot lesions of sizes 13, 37, and 13 mm of contrast 4 : 1 and one spherical cold lesion of 37 mm diameter. These lesions are positioned radially at 7 cm from the center along the vertical and horizontal axes of the off-center axial planes.

8.6.3 Quantitative measurements

To evaluate the performance of the LMEM and MGMF-LMEM algorithms, the contrast recovery coefficient (*CRC*) as defined in the NEMA NU 2007 protocol [147] and the signal to noise ratio (SNR) were used. The *CRC* is calculated with

$$CRC_H = \frac{C_H/C_B - 1}{A_H/A_B - 1} \times 100 \tag{8.10}$$

$$CRC_C = 1 - C_C/C_B \times 100 \tag{8.11}$$

where C is the average number of counts in a particular region of interest (ROI) identified by the subscripts H, C and B referring to hot, cold and background respectively. The diameter of each ROI is equal to the diameter of the spheres. The average was calculated over four distinct realisations of Monte Carlo simulations. A_H and A_B are the theoretical (simulated) activities in the hot and background regions respectively. For each lesion, eight equally sized ROIs were defined in different parts of the background and C_B therefore correspond to the average number of counts over 32 ROIs (four realisations). The standard deviation σ_B of counts in these 32 background regions was used to compute the SNR for each lesion as

$$SNR = \frac{C_H - C_B}{\sigma_B}. \tag{8.12}$$

The relative mean square error (RMSE) between reconstructed images I_{recon} and the true distributions I_{ref} was also used as a figure of merit. The RMSE is calculated according to

$$\text{RMSE}(I_{recon}, I_{ref}) = \sqrt{\frac{\sum_{j \in FOV} (I_{recon}(j) - I_{ref}(j))^2}{\sum_{j \in FOV} I_{ref}(j)^2}} \tag{8.13}$$

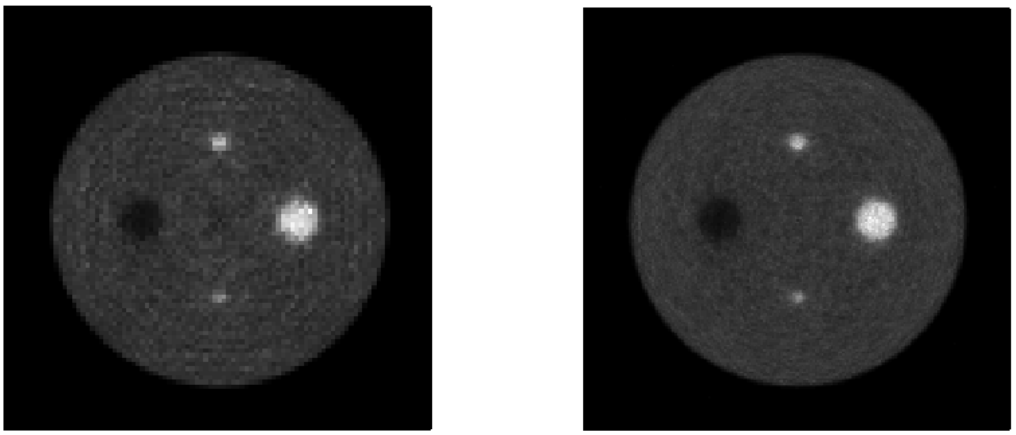
where FOV means the transverse field-of-view.

8.7 Results and discussion

8.7.1 Stopping criterion validation

To validate the proposed stopping criterion (SC) defined in section 8.4.1, the SNR, the CRC and the value of SC as a function of the iteration number were plotted in Fig. 8.3 for the reconstructed images shown in Fig. 9.5. The SNR sharply reaches a maximum and decreases, more rapidly for large lesions. The CRC increases rapidly for the first few iterations and then at a reduced rate as the iterations progress.

Since the maximal SNR occurs when the CRC is relatively low, the optimal number of iterations is a trade-off between noise and quantification accuracy. The SC varies over a relatively



a) b)

Figure 8.2 – Reconstructed images of the contrast phantom (off-center axial plane) using the LMEM algorithm for 20 iterations, 65 millions of events and a) $94 \times 94 \times 57$ voxels of $4 \times 4 \times 3.15$ mm^3 using two rays per detector pair, and b) $188 \times 188 \times 57$ voxels of $2 \times 2 \times 3.15$ mm^3 voxels using three rays per detector pair.

wide range for the first few iterations and then tends slowly to 1 as the number of iteration increases. So SC can be used as the stopping criterion and its value is a trade-off between convergence and SNR. The iteration process was stopped for the current grid when the SC belonged to the interval $[1-L, 1+L]$ where L is a preset value which determines the trade-off between convergence and SNR. Since the convergence of the SC to 1 is faster for the $188 \times 188 \times 57$ array than for the $94 \times 94 \times 57$ array, and since the MGMF-LMEM algorithm recovers only the low frequency components for a coarse grid, the value of L chosen for this case was larger than the one chosen for finer grids. This can contribute to faster convergence with minimal impact on the final image. In this work, values of $L = 0.01$ and $L = 0.02$ were used for the $188 \times 188 \times 57$ and $94 \times 94 \times 57$ grids respectively.

8.7.2 Computation time

The execution times to perform a GPU-based interpolation from a $94 \times 94 \times 57$ array to a $188 \times 188 \times 57$ array were 1.2, 1.2 and 1.4 ms for the nearest-neighbor, the bilinear and the B-spline methods respectively. Since the interpolation is performed only once at each phase, the

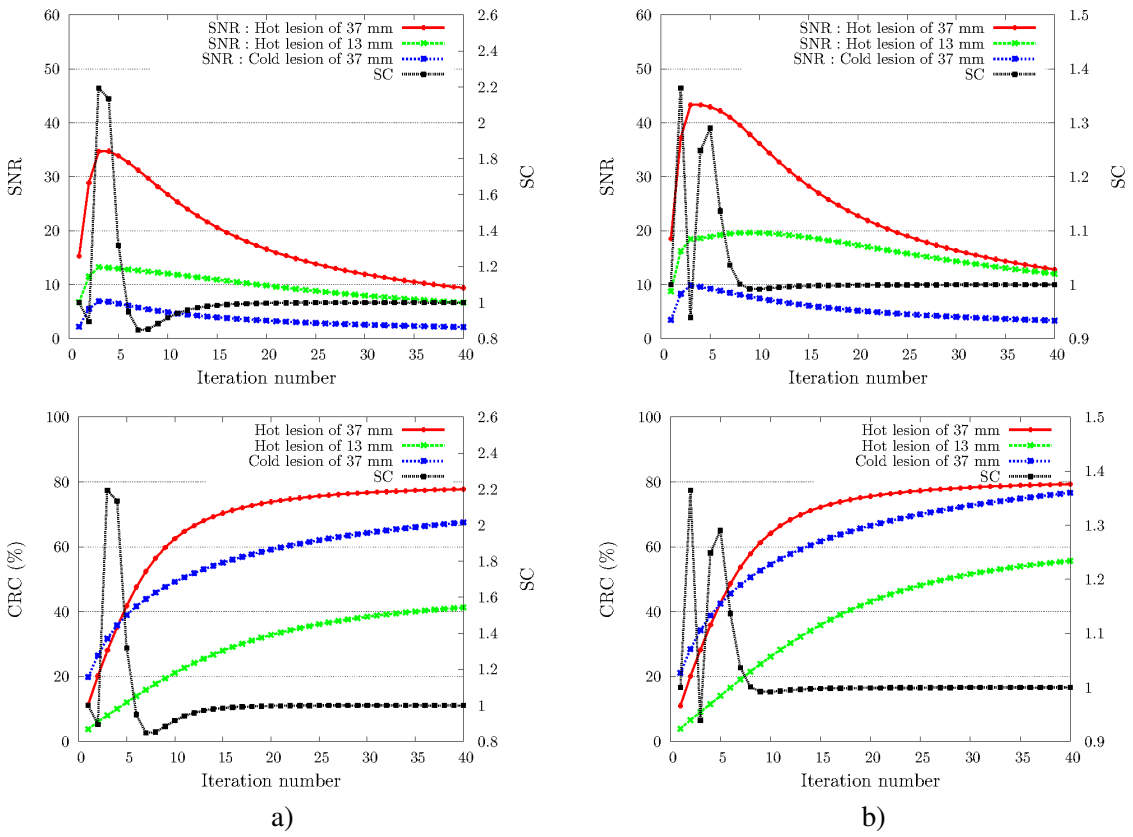


Figure 8.3 – SNR (upper row) and CRC (bottom row) as a function of the number of iterations calculated for the off-center lesions using the LMEM algorithm for a) $94 \times 94 \times 57$ voxels of $4 \times 4 \times 3.15$ mm^3 and b) $188 \times 188 \times 57$ voxels of $2 \times 2 \times 3.15$ mm^3. The value of the stopping criterion (SC) is also shown on the right scale of the graphs.

interpolation method has a negligible impact on the total reconstruction time. For this reason, the B-spline interpolation is the preferred method as it leads to the best results as explained in Sec. 8.7.3.

The time required to apply the 3D Gaussian filter to the reconstructed image at the end of each phase was 87 and 350 ms for the $Grid_0$ and $Grid_1$ respectively. Corresponding times for the Frobenius norm were 0.15 and 0.22 ms. These times are negligible in comparison to the computation time for one iteration at each phase of the MGMF-LMEM, as reported in Tab. 8.I. The computation time of the MGMF-LMEM algorithm during each phase is the same as for the LMEM algorithm for the grid corresponding to this phase.

These computation times for the MGMF-LMEM algorithm allow for the fulfillment of the

Tableau 8.I – Execution times for computation of the sensitivity matrices and for one iteration of the LMEM algorithm using multiple rays per detectors pair and one million of events. The computation time of the MGMF-LMEM during each phase is the same as for the LMEM algorithm for the grid corresponding to this phase.

$94 \times 94 \times 57$ matrix of $4 \times 4 \times 3.15$ mm^3		$188 \times 188 \times 57$ matrix of $2 \times 2 \times 3.15$ mm^3	
2 tangential $\times$ 2 axial sub-LORs		3 tangential $\times$ 2 axial sub-LORs	
Senstivity matrix (s)	**LMEM (s)**	**Senstivity matrix (s)**	**LMEM (s)**
1.9	0.35	9.8	1.1

stopping criterion during the current phase, so that the next reconstruction step starts at the beginning of the next acquisition phase without delay. For example, simulation results reported in table 8.II show that the computation time of the MGMF-LMEM algorithm during the phase 1 for a $94 \times 94 \times 57$ matrix, one iteration and 27 millions of events is approximately 9.5 s (the computation time is proportional to the number of events). Therefore, 63 iterations of the MGMF-LMEM algorithm can be executed during the phase 1 of a 10 minutes acquisition, which is more than the 18 iterations required by the stopping criteria. The same argument holds for the $188 \times 188 \times 57$ matrix, with convergence achieved at iteration 8 while 25 could be processed in a phase 2 lasting 20 minutes and containing 47 millions of events.

For low count statistics acquired during short acquisitions, other simulation results reported in table 8.II show that 16 iterations are required to achieve the convergence of the MGMF-LMEM algorithm during phase 1 for 9 millions of events and a $94 \times 94 \times 57$ grid while 12 iterations are required during phase 2 for 17 millions of events and a $188 \times 188 \times 57$ grid. The reconstruction times were 52 seconds during phase 1 and 225 seconds during phase 2. This computation time allows to reach convergence before the next phase starts for bed position duration of up to 5 minutes for typical activity levels.

Tableau 8.II – Number of iteration and time required to achieve the convergence for the MGMF-LMEM algorithm during phase 1 and phase 2.

	Phase 1		Phase 2	
Grid	$94 \times 94 \times 57$ matrix of $4 \times 4 \times 3.15$ mm^3		$188 \times 188 \times 57$ matrix of $2 \times 2 \times 3.15$ mm^3	
Events number (million)	9	27	17	47
Iterations number	16	18	12	8
Time (s)	52	175	225	418

The challenge of MGMF-LMEM is to ensure the convergence of the EM algorithm at each

phase. This becomes quite challenging for short acquisitions with high activity. In this case, the solution is to use only one grid (the finest grid $188 \times 188 \times 57$ voxels) and two frames $[0, T_1]$ and $[T_1, T_{aq}]$ over the total acquisition time T_{aq} for one bed position. The first execution of LMEM will start at T_1 using the acquired event during interval $[0, T_1]$ and will stop when the convergence criterion is satisfied and before the acquisition is finished at T_{aq}. The second computation of LMEM will start at T_{aq} using all acquired events and the previously reconstructed image as initial image. To ensure the convergence of first execution of LMEM during the interval $[T_1, T_{aq}]$, the value of T_1 will be chosen according to the injected activity. As the injected activity increases, the value of T_1 will be decreased in order to increase the calculation interval $[T_1, T_{aq}]$ and to reduce the number of events that will be used to compute the LMEM algorithm during this interval. Determining the optimal value of T_1 for each injected activity needs to be done in clinical use. But the best choice to overcome the problem of convergence is to use multi-GPUs to speed up the convergence of LMEM at each phase [44], as well as the combination of both strategies.

The challenge of MGMF-LMEM is to ensure the convergence of the EM algorithm at each phase. This becomes quite challenging for short acquisitions with high activity. In this case, the solution is to use only one grid (the finest grid $188 \times 188 \times 57$ voxels) and two frames $[0, T_1]$ and $[T_1, T_{aq}]$ over the total acquisition time T_{aq} for one bed position. The first execution of LMEM will start at T_1 using the acquired event during interval $[0, T_1]$ and will stop when the convergence criterion is satisfied and before the acquisition is finished at T_{aq}. The second computation of LMEM will start at T_{aq} using all acquired events and the previously reconstructed image as initial image. To ensure the convergence of first execution of LMEM during the interval $[T_1, T_{aq}]$, the value of T_1 will be chosen according to the injected activity. As the injected activity increases, the value of T_1 will be decreased in order to increase the calculation interval $[T_1, T_{aq}]$ and to reduce the number of events that will be used to compute the LMEM algorithm during this interval. Determining the optimal value of T_1 for each injected activity needs to be done in clinical use. But the best choice to overcome the problem of convergence for short acquisitions with high activity is to use multi-GPUs to speed up the convergence of LMEM at each phase [44], as well as the combination of both strategies.

Finally, the computation time of the sensitivity matrices presented in Tab. 8.I is less than 10 s for the $188 \times 188 \times 57$ grid and less than 2 s for the $94 \times 94 \times 57$ grid. Therefore, the computation of the sensitivity matrices is completed during the phase 0, *i.e.*, before the execution of the MGMF-LMEM algorithm is started.

8.7.3 Performance evaluation of the MGMF-LMEM algorithm

In the following section, the computation time is calculated from the moment the data acquisition was stopped (T_{aq}) and the number of iterations of the MGMF-LMEM algorithm correspond to iterations from this moment, which is the beginning of the last phase of the algorithm. From this moment, the computation time for one iteration of the MGMF-LMEM algorithm is the same as the computation time for one iteration of LMEM algorithm, with both algorithms using all acquired events and the same grid.

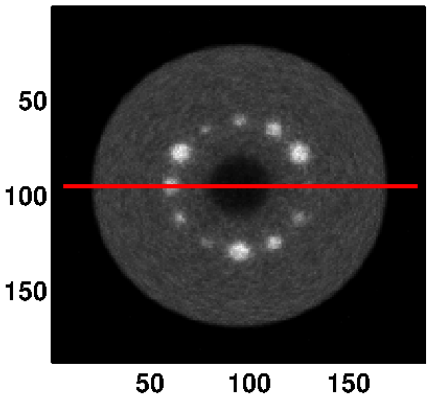
Tableau 8.III – Number of iteration required to achieve 85% of maximum CRC for simulated lesions using 65 millions events for the LMEM and the MGMF-LMEM algorithms.

	Lesion			
	10 mm hot	17 mm hot	22 mm hot	50 mm cold
LMEM (iterations)	35	28	19	20
MGMF-LMEM (iterations)	10	4	4	2
Acceleration of convergence	3.5	7	4.7	10

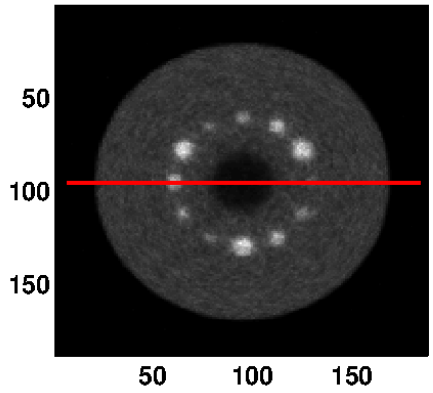
To compare the LMEM and MGMF-LMEM algorithms, the reconstruction of the contrast phantom was performed using SC=0.995 (L=0.005) in each case. The convergence of LMEM was reached after 22 iterations while 7 iterations were required for MGMF-LMEM no matter which interpolation method was used. Images obtained were very similar. The RMSE between the central slice obtained with LMEM and MGMF-LMEM were 0.0176, 0.0099 and 0.0169 respectively for the nearest-neighbour, bilinear and cubic B-spline interpolation methods.

Furthermore, figure 8.5 shows the performance of the LMEM and MGMF-LMEM algorithms in terms of CRC for various lesion sizes for the three interpolation schemes used in this study, with a data set of 65 millions of events. While both algorithms ultimately converge to similar CRC, the MGMF-LMEM version reaches high CRC values with fewer iterations than the LMEM version. Taking 85% of maximum CRC as a reference, the MGMF-LMEM algorithm reaches this threshold with at least three times less iterations than the LMEM algorithm for all lesions sizes as Tab. 8.III and Fig. 8.5 show. This figure also shows that the interpolation method used has a limited impact on the CRC, except for the 10 mm hot lesion where the cubic B-spline leads to higher CRC values.

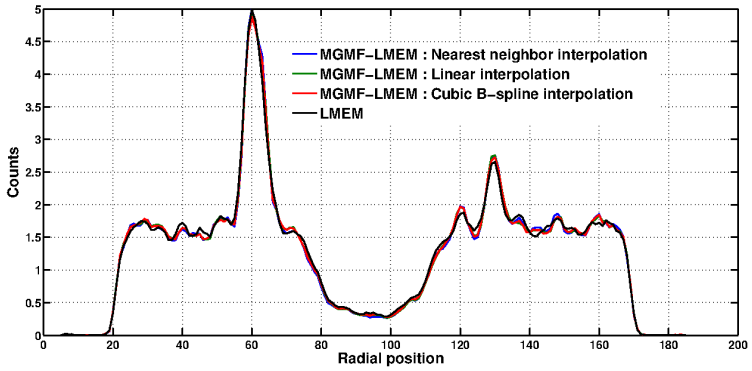
Figure 8.6 a) shows the RMSE relative to the reference image of the central axial plane of the contrast phantom, for the LMEM and the MGMF-LMEM algorithms using three interpolation methods. The minimum of RMSE is reached at first iteration for MGMF-LMEM and at itera-



a) 22 iterations LMEM



b) 7 iterations MGMF-LMEM with linear interpolation

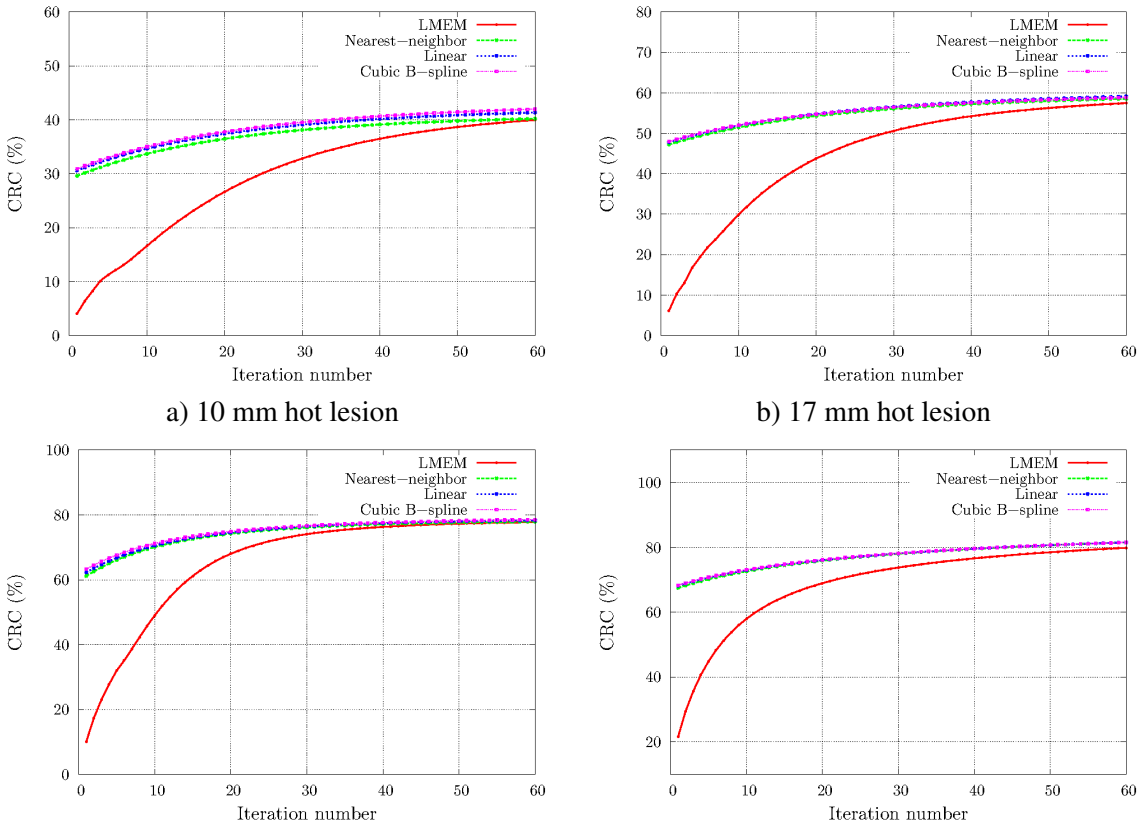


c

Figure 8.4 – Reconstructed images of the central axial plane of the contrast phantom computed for 65 millions of events for SC=0.995 (L=0,005) : a) 22 iterations of LMEM, b) 7 iterations of MGMF-LMEM with linear interpolation, and c) corresponding cut-views

tion 15 for LMEM, again pointing towards faster convergence of the MGMF-LMEM algorithm. According to this criterion, linear and cubic B-spline interpolations perform equally well while nearest-neighbor interpolation leads to slightly worse results.

Otherwise, figure 8.6 b) shows that the MGMF-LMEM algorithm has a poorer behavior with respect to noise compared to LMEM. Previous iterations of MGMF-LMEM during the acquisition accelerate the convergence with respect to LMEM but also increase the noise. In addition, figures 8.5 and 8.6 show that cubic B-spline provides a slightly better CRC and SNR performance compared to nearest-neighbor and bilinear interpolations. According to figure 7, the bilinear interpolation scheme seems to provide better convergence. This may be explained by



a) 10 mm hot lesion b) 17 mm hot lesion

c) 22 mm hot lesion d) 50 mm cold lesion

Figure 8.5 – CRC for lesions on the central axial plane of the contrast phantom using the MGMF-LMEM algorithm and different interpolation methods (65 millions of events). a) 10 mm hot lesion, b) 17 mm hot lesion , c) 22 mm hot lesion, d) 50 mm cold lesion.

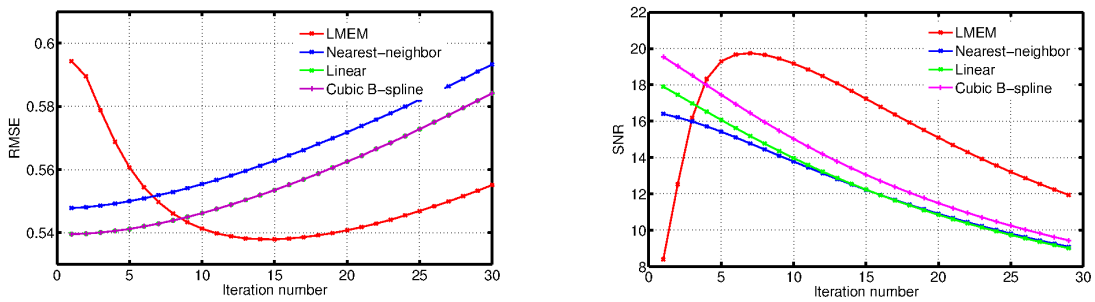
cubic B-spline preserving gradients better than bilinear interpolation

Figure 8.7 also demonstrates that the asymptotic behavior of the MGMF-LMEM algorithm using linear and cubic B-spline interpolation is the same as the LMEM algorithm.

As summary, the results show that :

– The bilinear and cubic B-spline interpolation methods provide clearly the best performances compared to the nearest-neighbour method (Fig. 8.5, 8.6 and 8.7). The cubic B-spline is slightly better than the bilinear method (Fig. 8.5 and 8.6). Since the computation time of these two last methods is almost the same, the best choice is the cubic B-spline interpolation method.

– The two algorithms LMEM and MGMF-LMEM converge to the same CRC (Fig. 8.4 and



a) b)

Figure 8.6 – a) RMSE versus iteration of the reconstructed central axial of the contrast phantom plane relative to the reference image, and b) SNR versus iteration number for 22 mm hot lesion. RMSE and SNR are computed using 65 millions of events with the LMEM and MGMF-LMEM algorithms for nearest-neighbor, bilinear, and cubic B-spline interpolation kernels.

8.5), reach the same minimum for RMSE (Fig. 8.6) and the same log-likelihood (Fig. 8.7). Thus the two algorithms converge to the same solution. Furthermore, the MGMF-LMEM version converges at least three times faster than LMEM algorithm (Fig. 8.4 and 8.5, and Tab. 8.III). However, LMEM has a better behaviour with respect to SNR and RMSE (Fig. 8.6).

– The MGMF-LMEM computation time on the GPU was 1.1 s/million of event per iteration (Tab. 8.I). As convergence was obtained in four iterations of the MGMF-LMEM algorithm for the 17 and 22 mm lesions (Tab. 8.III and Fig. 8.5), convergence can be reached in approximately five minutes for an acquisition of 65 millions of events. Furthermore, since the MGMF-LMEM algorithm provides 75% of maximum CRC and minimum RMSE at the first iteration (Fig. 8.5 and 8.6), MGMF-LMEM can be used as a one-pass algorithm to perform real-time reconstructions for low count acquisitions such as gated studies. The computation time for one iteration and 60 millions of events is approximately 66 s.

8.8 Conclusion and future work

In this work, the multigrid/multiframe approach to the expectation-maximization algorithm was extended to list-mode to overcome obstacles that limit the acceleration of this type of acquisition on the GPU. A stopping criterion that is computationally more efficient on GPU also was introduced.The MGMF-LMEM algorithm was tested using Monte Carlo simulated data and the

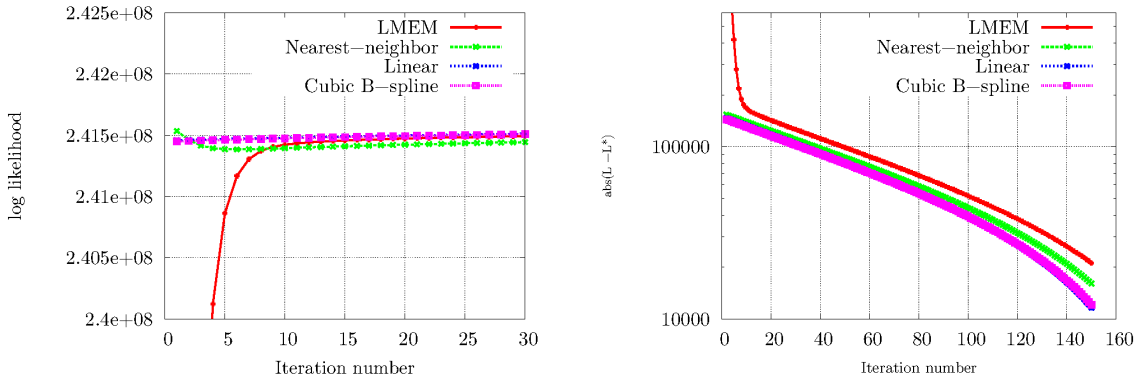


Figure 8.7 – a) log-likelihood of the reconstructed central axial plane of the contrast phantom relative to the reference image, as a function of the number of iterations, for the three interpolation methods studied. b) Residual difference $|L-L^*|$, where L is the log likelihood of MGMF-LMEM algorithm for different interpolation methods and L^* is the log likelihood of LMEM algorithm at iteration 200.

results obtained show that it converges to the same solution as LMEM but reaches the solution at least three times faster. The contrast recovery obtained at the first iteration was larger than 75% (Fig. 8.5). The GPU-based MGMF-LMEM algorithm processed data at a rate close to one million of events per second per iteration, and lets envision near real-time reconstructions for large acquisitions or low-count acquisitions such as gated studies. The interpolation method used had negligible impact on the computation time and on the accuracy of the results.

Even if these results are promising, the evaluation of this implementation using clinical data will be required. The optimization of the number of grids and the number of events per grid will also be explored. This work will be done after the integration of random and scatter corrections into the algorithm. These corrections will ideally be implemented to avoid a computational bottleneck in the proposed reconstruction strategy. In order to do so, a second GPU board dedicated to corrections could be used as proposed by Wang [234]. Scatter corrections will be estimated using the fast GPU-oriented Monte Carlo platform developed by our group [75]. The detector's point spread functions (PSFs) will also be included and might lead to improved CRC [227].

As demonstrated by Ho *et al.* [154], the likelihood function should be modified to ensure that it is increasing monotonically when the grid is changed. Although this was not done here, results suggest that the proposed method works as expected. But, the influence of an adapted likelihood function needs also to be explored in future work.

Acknowledgements

This work was supported by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT) and by the Natural Sciences and Engineering Research Council of Canada (NSERC). NVIDIA Corporation kindly donated material to conduct this study.

CHAPITRE 9

IMPLANTATION SUR GPU DE L'ALGORITHME HAUTE RÉSOLUTION : 3D OSEM PAR PONDÉRATION DES LIGNES DE RÉPONSE POUR LA CORRECTION DE L'ATTÉNUATION

9.1 Abstract

Purpose : The attenuation-weighted line-of-response OSEM (AW-LOR-OSEM) algorithm allows a PET image reconstruction from sinograms without any data compression (span=1, mashing=1) and incorporates the attenuation and normalization corrections in the sensitivity matrices as weight factors. The main objective of this work is to accelerate the computation of this algorithm for modern PET scanners as the Philips Gemini GXL by its implementation on modern GPU devices.

Methods : We implemented the AW-LOR-OSEM algorithm on the NVIDIA Tesla C2050 GPU. The system matrix (SM) is computed using the multi-ray tracing Siddon algorithm. We compared two strategies in this implementation : first SM was pre-calculated using 6 rays per detector pair in the tangential direction and 2 rays in the axial direction and stored (1.4 Gigabytes). Secondly, the SM was calculated on-the-fly using 3 rays per detector pair in the tangential direction and 2 rays in the axial direction. To reduce the computation time, the symmetries of the scanner were exploited and the matrices data were reorganized to allow coalesced GPU memory access. These implementations were validated using Monte Carlo simulated data with the GATE package.

Results : The reconstruction was computed for a 188x188x57 array (FOV= 376 mm, 2x2x3.15 mm^3 voxel size). For the implementation using pre-calculated SM, the time to compute the sensivity matrices is 22 seconds and the time to compute the LOR-OSEM algorithm for 1 iteration, 11 subsets, and 112 million coincidences is 18 seconds. For the implementation using the calculated SM on-the-fly, the times are 10 and 8 seconds for, respectively, the computation of sensitivity matrices and LOR-OSEM algorithm. The sensitivity matrices are only computed in the first iteration and kept in the GPU memory for the other iterations for both implementations.

Conclusions : The AW-LOR-OSEM algorithm was successfully implemented on a Tesla C2050 GPU for a PET system that has about 85 million LORs. The results show that the compu-

tation efficiency is about twice better for the implementation using calculated SM on-the-fly than the implementation using pre-calculated SM. The reported reconstruction times are compatible with a clinical use for both strategies. New devices such as the Tesla K20X GPU and Tesla K40X are equipped with large memory (respectively 6 and 12 GB), which will allow the use of a very large SM pre-calculated with more precision. Future works are to explore the compute of the system matrix using Monte Carlo simulation and to implement the random and scatter events corrections, which are time-consuming.

9.2 Résumé

Objectif : L'algorithme 3D OSEM par pondération des lignes de réponse pour la correction de l'atténuation (AW-LOR-OSEM : *attenuation-weighted line-of-response* 3D OSEM) estime la distribution 3D du traceur à partir des données de projections en sinogramme sans compression (*span=1,mashing=1*) et effectue la correction d'atténuation et de normalisation durant la reconstruction en intégrant les coefficients de corrections dans la matrice système (MS) pour préserver la nature stochastique des données d'acquisition ([96]). L'objectif de ce travail est d'accélérer la reconstruction sur GPU de cet algorithme haute résolution AW-LOR-OSEM pour les systèmes TEP modernes ayant un nombre de LORs de l'ordre de 85 millions comme le Gemini GXL de Philips.

Méthodologie : Deux implantations de AW-LOR-OSEM ont été développées sur GPU Tesla C2050. La première se base une MS précalculée par la méthode multi-trajectoires de Siddon en utilisant 6 rayons par LOR dans la direction tangentielle et 2 dans la direction axiale. La deuxième implantation détermine en temps réel les coefficients de MS en utilisant 3 rayons par LOR dans la direction tangentielle et 2 dans la direction axiale. Afin d'accélérer le temps de calcul, les symétries du système TEP ont été exploitées et des stratégies ont été développées pour assurer les conditions de la coalescence d'accès à la mémoire globale de GPU. Les données TEP pour valider les deux implantations ont été simulées par Monte Carlo en utilisant le logiciel GATE.

Résultats : Les reconstructions ont été effectuées sur une matrice de définition 188 x 188 x 57 voxels de taille 2x2x3.15 mm^3. Pour l'implantation utilisant une MS précalculée, le temps de calcul pour déterminer les matrices de sensibilité est de 22 secondes et le temps nécessaire pour exécuter une itération de l'algorithme LOR-OSEM pour 11 sous-ensembles et 112 millions d'événements est de 18 secondes. Pour la deuxième implantation, qui calcule la MS en temps

de réel, ces temps sont de 10 et 8 pour respectivement déterminer les matrices de sensibilité et exécuter une itération de LOR-OSEM.

Conclusion : Les temps d'exécution obtenus permettent l'utilisation routinière en clinique de l'algorithme AW-LOR-OSEM. Par ailleurs, les performances en temps de calcul obtenus pour l'implantation qui détermine la MS en temps réel est le double de celle qui utilise une MS pré-calculée. Malgré ces résultats, le temps d'exécution de la deuxième implantation est très rapide et elle doit être privilégiée en utilisation clinique vue sa meilleure exactitude de quantification. De plus, les dispositifs GPU modernes tels que Tesla K20X GPU et Tesla K40X sont équipés de mémoire très étendue (respectivement 6 et 12 GB) permettant l'utilisation d'une MS très grande qui sera pré-calculée avec plus d'exactitude en utilisant des méthodes plus complexes. Le futur travail est donc d'utiliser une MS pré-calculée par Monte Carlo sur ces nouveaux dispositifs et d'intégrer à l'implantation les fonctions de correction du diffusé et des événements fortuits.

9.3 Introduction

Les algorithmes stochastiques itératifs de reconstruction à partir des données d'acquisitions stockées sous forme des sinogrammes sont devenus les méthodes standards de quantification en TEP. Ces méthodes modélisent implicitement la nature stochastique des données d'acquisition. Ils considèrent que ces données suivent une distribution de Poisson, et déterminent par itération la distribution du traceur qui maximise la vraisemblance entre les mesures et les données de projection estimées. Les méthodes stochastiques offrent un bon RSB en comparaison aux algorithmes analytiques déterministes tels que la rétroprojection filtrée, qui était la méthode la plus utilisée auparavant.

L'algorithme stochastique itératif OSEM proposé par Hudson et Larkin [84] est largement le plus utilisé actuellement pour la reconstruction des images TEP. Son principe consiste à corriger l'image estimée en utilisant un sous-ensemble des sinogrammes à chaque itération. Cet algorithme a été développé pour accélérer l'algorithme de référence (*gold standard*) MLEM proposé par Shepp et Vardi [205]. Ce dernier utilise l'ensemble des données de projection pour mettre à jour l'image à chaque itération et c'est un algorithme qui converge lentement. Cependant, contrairement à MLEM, OSEM ne converge pas généralement vers la solution de maximum de vraisemblance [55, 84].

Les développements technologiques des deux dernières décennies ont permis d'augmenter le

nombre de détecteurs dans les systèmes TEP modernes. Par conséquent, le nombre de LORs au niveau de ces systèmes a augmenté exponentiellement. Par exemple, le système TEP de Gemini GXL de Philips présente 85 millions LORs alors que l'ECAT HRT de Siemens offre 4.5 milliards de LORs. Ce développement a amélioré considérablement la résolution des images estimées et la sensibilité des systèmes TEP. Mais il a aussi augmenté énormément le temps de calcul des algorithmes de reconstruction itératifs, ce qui compromet l'utilisation en clinique de routine de l'algorithme 3D OSEM. En effet, malgré le développement rapide de la puissance des ordinateurs, la puissance de calcul disponible est toujours en retard par rapport aux besoins créés par l'augmentation du nombre des données d'acquisition et l'utilisation des méthodes de reconstruction plus sophistiquées.

Comme il a été expliqué au paragraphe 4.1, deux approches ont été développées pour accélérer l'exécution de 3D OSEM. La première méthode consiste à redistribuer les données 3D en une pile de projections 2D, et puis à estimer à partir de ces projections les coupes axiales 2D qui sont indépendantes les unes des autres. La seconde approche compresse les données de projection en fusionnant de nombreux sinogrammes voisins dans un seul sinogramme pour réduire la taille des données (voir la section 2.6.3 pour plus de détails). Cependant, il a été montré que ces deux approches réduisent la qualité de la quantification [241]. Le mode liste est aussi utilisé pour accélérer l'algorithme 3D OSEM dans le cas des acquisitions dynamiques ou le nombre des événements est inférieur au nombre des LORs [183, 189]. Mais, le gain en temps de calcul est limité. En effet, comme il a été expliqué aux chapitres 5 et 6, d'une part, le mode liste ne permet pas d'utiliser les symétries pour accélerer la reconstruction, et d'autre part, pour ce mode, le calcul de la matrice de sensibilité, les corrections du diffusé et celles des événements fortuites nécessistent un temps très long [35, 127, 175].

Par ailleurs, pour accélérer la reconstruction, le calcul de la MS se fait généralement en temps réel durant la reconstruction en utilisant la méthode analytique simple multi-trajectoires de Siddon [77, 209, 254] (voir la section 3.2.2). Cependant, l'utilisation d'une MS précalculée permet de bien modéliser le système d'acquisition et donc d'améliorer la quantification TEP [253].

L'algorithme 3D OSEM par pondération des lignes de réponse pour la correction de l'atténuation (AW-LOR-OSEM : *attenuation-weighted line-of-response* 3D OSEM) est un algorithme qui estime la distribution 3D du traceur à partir des données de projections en sinogramme sans compression (*span=1,mashing=1*) et qui effectue la correction d'atténuation et de normalisation

durant la reconstruction en intégrant les coefficients de corrections dans MS pour préserver la nature stochastique des données d'acquisition [96]. L'objectif de ce travail est : 1) accélérer la reconstruction sur GPU de l'algorithme haute résolution AW-LOR-OSEM, 2) comparer la performance de calcul sur GPU de deux implantations de l'algorithme AW-LOR-OSEM : une qui utilise une MS précalculée et l'autre qui détermine en temps réel les coefficients de MS. Ainsi, nous avons implanté sur GPU Tesla C2050 l'algorithme AW-LOR-OSEM pour le système de Philips Gemini GXL pour les deux approches de calcul de MS. L'accent a été mis surtout dans ce chapitre pour la version de l'algorithme utilisant une MS précalculée car elle présente des défis sur la gestion et la coalescence de la mémoire globale. Pour l'autre version se basant sur une MS calculée en temps de réel, la problématique de la coalescence de la matrice globale est similaire à celle de calcul de la matrice de sensibilité et qui était détaillée au chapitre 6.

9.4 Principe de l'algorithme AW-LOR-OSEM

L'algorithme 3D AW-LOR-OSEM consiste à : i) enregistrer sans compression les données de projection dans des sinogrammes. Chaque élément des sinogrammes est associé à une LOR et il contient le nombre d'événements détectés par la paire de détecteurs qui forme cette LOR, ii) diviser l'espace des sinogrammes $\mathbf{S}$ en un nombre p de sous-ensembles (*subsets*) $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_p$ qui forment une partition, où p est l'ordre de la partition et iii) estimer l'image selon l'équation :

$$Gr_j^m = \sum_{b \in S_l} p_{b,j} \frac{y(b)}{\sum_{i=1}^{J} p_{b,i} \lambda_i^m + \frac{s_b + r_b}{a_b \varepsilon_b}} \tag{9.1}$$

$$\lambda_j^{m+1} = \frac{\lambda_j^m}{N_j} Gr_j^m \tag{9.2}$$

$$N_{j,l} = \sum_{b \in S_l} p_{b,j} a_b \varepsilon_b \tag{9.3}$$

où :

- Gr_j^m et λ_j^m sont les valeurs du voxel j pour respectivement la matrice gradient Gr et la matrice image de la distribution du traceur λ estimées lors de l'exécution de la sous-itération m, S_l est le sous-ensemble qui correspond à $l = m$ modulo p, et $y(b)$ est le nombre des événements détectés le long du LOR b.
- s_b et r_b sont respectivement les valeurs moyennes du diffusé et des événements fortuits qui ont lieu sur la LOR b.

- Le coefficient $p_{i,j}$ est l'élément de MS associé à la LOR i et au voxel j. Il représente la probabilité qu'un événement qui a lieu au niveau de la LOR j soit détecté par la paire des détecteurs i.
- $N_{j,l}$ sont les éléments j des matrices de sensibilité N_l pour chaque sous-ensemble S_l. Elles corrigent pour la variation de la sensibilité entre les différentes LORs, et qui est causée par l'atténuation dans le patient et par la non-uniformité de l'efficacité de détection entre les différentes paires de détecteurs.

La détermination de $N_{j,l}$ nécessite de calculer pour chaque LOR i de S_l : 1) le facteur d'atténuation a_i par la projection de la matrice M_μ des coefficients d'atténuation des tissus à 511 keV estimée à partir d'une acquisition TDM par $a_i = exp(-l\sum_{j=1}^{j=J} p_{ij}\mu_j)$, où l est la longueur de la trajectoire dans le voxel j, et 2) la rétroprojection dans l'espace image du produit $w_i = \varepsilon_i \mathrm{x} a_i$, où ε_i est le facteur de normalisation qui tient compte de la variation de l'efficacité de détection entre les différentes paires de détecteurs [51, 153].

Par ailleurs, afin d'assurer une convergence rapide de l'algorithme, le choix des sous-ensembles est fait pour permettre une séparation maximale des LORs dans chaque plan transversal. De ce fait, la probabilité de détecter un événement provenant d'un voxel sera la même pour toutes les sous-ensembles[84]. Ainsi, chaque sous-ensembles i est choisi pour contenir les LOR(z,θ,r,ϕ) dont l'indice de vue ϕ vérifie la relation ϕ modulo $p = i$, avec p est le nombre de sous-ensembles sélectionnés. z,θ,r,ϕ sont respectivement les indices des positions axiales, de l'angle azimutal, de la position radiale sur le plan de coupe et de la l'angle polaire dans ce plan.

9.4.1 Symétries

Le système d'acquisition Gemini GXL est un système cylindrique de 29 anneaux dont chacun est constitué de 616 détecteurs (tableau 9.I). Comme il a été expliqué au paragraphe 7.3.3, ce système présente 8 symétries transversales (figure 7.1) et des symétries axiales de translation et de réflexion (figure 9.1) qui sont exploitées pour réduire la taille de la MS et accélérer la reconstruction. Le nombre de symétries axiales se traduit par une réduction d'un facteur 29 de la taille de la MS. En effet, si nous choisissons une matrice de reconstruction dont la taille des voxels dans la direction Z est un sous-multiple de la distance entre les centres de deux couronnes voisines qui est de 4,3 mm, alors les LORs ayant le même angle azimutal indexé par $\theta_{i,j} = j - i$ présentent $29 - \theta_{i,j}$ symétries axiales de translation, où i et j sont les indices des deux couronnes. De plus, les deux LOR$(z_{i,j},\theta_{i,j},r,\phi)$ et LOR$(z_{j,i},\theta_{j,i},r,\phi)$ sont symétriques par réflexion. L'indice de la

position axiale $z_{i,j}$ est égale à $i+j-1$.

Pour les deux implantations ; la première qui utilise une MS précalculée et la deuxième qui détermine en temps réel les coefficients de MS, la définition de la matrice de reconstruction utilisée est de 188 x 188 x 57 voxels de taille 2 x 2 x 3.15 mm^3. Pour cette définition, comme le montre le tableau 9.I, la taille de la MS, est 688 To. En utilisant les 8 symétries transversales et les 29 symétries axiales, et en ne stockant que les éléments non nuls, la taille de la MS est réduite à 1.2 Go.

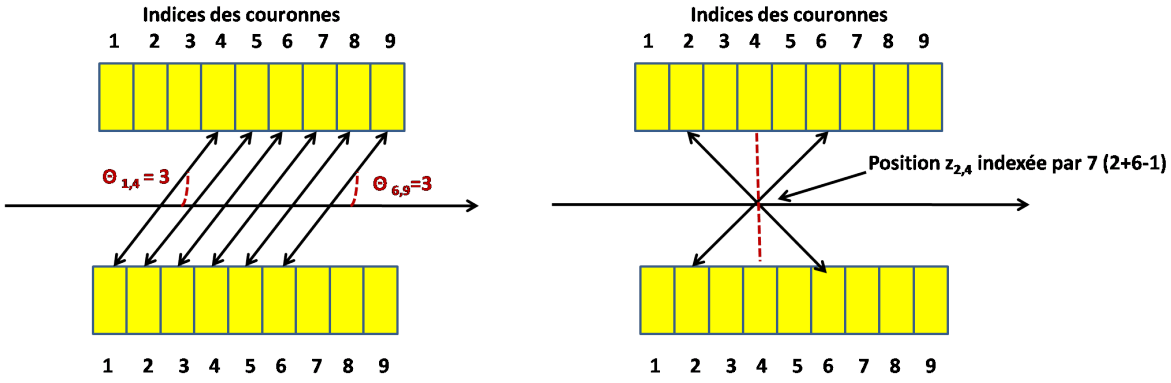


Figure 9.1 – Figure montrant a) les symétries axiales par translation, b) les symétries axiales par réflexion. Source : [253]

9.5 Implantation sur GPU

Deux implantations de l'algorithme AW-LOR-OSEM ont été réalisées sur GPU Tesla C2050 pour le système Philips Gemin GXL :

- Implantation 1 qui calcule les coefficients $p_{i,j}$ de MS au fur et à mesure de leur utilisation pour effectuer la projection des données de l'espace des images vers l'espace des projections, et la rétroprojection des données projection dans l'espace image. Ces coefficients sont calculés avec l'algorithme de multi-trajectoires de Siddon en utilisant 6 rayons par paire de détecteurs (3 dans la direction tangentielle et 2 dans la direction axiale).
- Implantation 2 qui utilise MS précalculée. Cette matrice a été aussi calculée par la méthode multi-trajectoires de Siddon. Puisque le nombre de rayons par LORs utilisé pour calculer MS n'affecte pas le temps d'exécution de l'algorithme AW-LOR-OSEM pour

Tableau 9.I – Système Philips Gemini GXL : taille de la matrice des projections (sinogrammes sans compression : *span*=1 et *mashing* =1) et taille de MS.

Matrice des projections	
Nombre de vues	308
Nombre de LORs par vue	330
Nombre de plan transversaux (sinogrammes)	841
Dimensions de la matrice de projection	85 479 240 (308 x 330 x 841) LORs
Taille de la matrice de projection	342 Mo
Matrice de reconstruction	
Dimensions de la matrice de reconstruction	188 x 188 x 57 voxels
Taille de la matrice de reconstruction	8.06 Mo
Taille des voxels	2 x 2 x 3.15. mm^3
Matrice système	
Dimensions de la MS (éléments)	1.722 10^{14}(188 x188x 57 x 85 479 240)
Taille de la MS (sans aucune compression)	688 To
Nombre de symétries transversales	8
Nombre de symétries axiales	29
Taille de la MS compressées (symétries et éléments non nuls)	1.2 Go

cette dernière implantation, nous avons utilisé 2 et 6 rayons respectivement dans la direction axiale et tangentielle pour permettre un bon échantillonnage de la matrice de reconstruction afin d'augmenter l'exactitude de la MS précalculée. En effet, dans la direction l'axiale, la taille des voxels (3.15 mm) est la moitie de la distance entre les centres de deux de détecteurs appartenant à deux couronnes voisines (6.30 mm) et, par conséquent, l'utilisation de 2 rayons par LOR dans la direction axiale est suffisant pour permettent un bon échantillonnage.

Comme le montre le pseudo-code 1 de l'algorithme AW-LOR-OSEM pour les deux implantations, 4 kernels ont été implémentés, entre autres : 1) le kernel *ComputeSensitivityMatrix* qui calcule les matrices de sensibilité N_l pour les différents sous-ensembles, 2) le kernel *ComputeGradient* qui calcule la matrice gradient *Gr* à chaque sous-itération, 3) le kernel *ImageUpdate* qui corrige la matrice image λ à chaque sous-itération, et 4) le kernel *GaussianFilter* qui applique un filtre gaussien à l'image estimée.

Algorithm 1 Pseudo-code de l'algorithme AW-LOR-OSEM.

```
__host__ AW-LOR-OSEM( )
{
for subset_number= 1 to number_of_subsets do
  ComputeSensitivityMatrix <<Basic_LORs>>(...)
end for
for iteration_number= 1 to number_iterations do
  for subset_number= 1 to number_of_subsets do
    ComputeGradient<<Basic_LORs>>(...)
    cudaThreadSynchronize() ;
    ImageUpdate<<voxels>>(...)
    cudaThreadSynchronize() ;
    GaussianFilter<<voxels>>(...)
  end for
end for
```

9.5.1 Calcul des matrices de sensibilité

Nous avons utilisé la même stratégie d'optimisation développée au chapitre 6 pour calculer la matrice de sensibilité pour l'algorithme LM-OSEM. Cependant, au lieu de calculer une seule matrice de sensibilité, nous calculons pour l'algorithme AW-LOR-OSEM une matrice de sensibilité pour chaque sous-ensemble en prenant en compte les LORs qui traversent le patient et qui appartiennent au sous-ensemble concernée. Afin d'assurer les conditions de coalescence, la matrice de normalisation est recopiée dans la mémoire globale sous forme d'une mémoire linéaire selon la structure présentée dans la figure 9.2. Lorsque l'exécution du kernel est finie, seules les matrices de sensibilités sont gardées dans la mémoire globale du GPU ; la matrice des coefficients d'atténuation M_μ et celle de la normalisation M_ε seront détruites pour libérer l'espace mémoire. Pour l'implantation 2 qui utilise une MS précalculée, cette dernière est aussi gardée dans la mémoire pour être utilisée par le kernel *ComputeGradient*. Puisque le chargement de la MS du disque dur vers la mémoire RAM CPU et son transfert vers la mémoire globale GPU sont longs, l'implantation permet de garder cette matrice dans la mémoire globale du GPU pour exécuter l'algorithme AW-LOR-OSEM pour les autres patients.

9.5.2 Calcul de la matrice gradient

Pour optimiser l'implémentation du Kernel *ComputeGradient* qui calcule la matrice gradient *Gr* à chaque sous-itération, nous avons adopté aussi la même stratégie utilisée pour calculer la

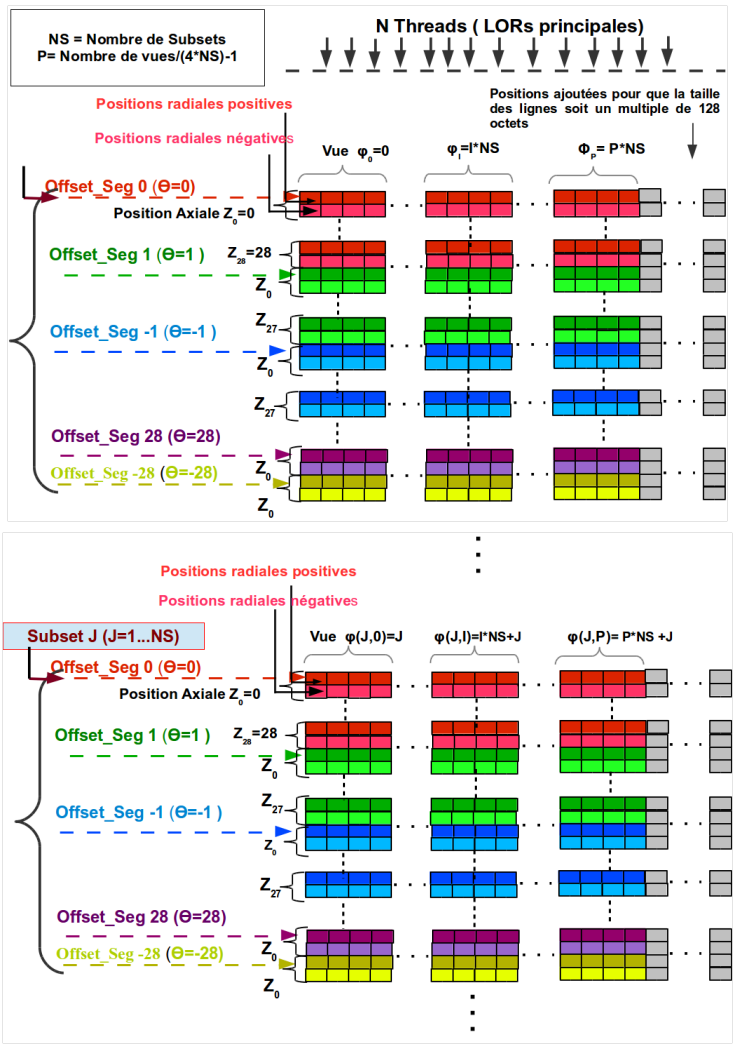


Figure 9.2 – Structure utililisée pour implémenter les données de projection dans la mémoire globale de GPU pour l'algorithme AW-LOR-OSEM.

matrice de sensibilité pour l'algorithme LM-OSEM et qui est détaillée au chapitre 6. En effet, comme le montre le pseudo code 1, chaque fil d'éxéction est associé à un LOR principal et : 1) effectue une projection de l'image actuelle dans l'espace projection le long de LOR principal et le long de ses symétries utilisées pour accélérer la reconstruction, 2) détermine les valeurs de correction pour le LOR principal et ses symétries en calculant le rapport entre les projections mesurées et les projections calculées et 3) réalise une rétroprojection de ces valeurs de corrections dans l'espace image. Cette stratégie d'effectuer la projection et la rétroprojection successivement avec le même fil d'éxéction permet d'éviter l'accès à la mémoire globale pour écrire et lire les valeurs des projections le long de tous les LORs. L'utilisation des symétries transversales seulement permet de limiter le nombre de registres nécessaires par fil d'exécution et par conséquent d'augmenter l'occupation des multiprocesseurs, comme il a été démontré au chapitre 6.

Pour assurer la coalescence de l'accès à la mémoire globale du GPU, la matrice image λ a été réorganisée et déclarée comme un tableau *3D CUDA ARRAY* de type *float4* que l'on associe à une texture 3D. Chaque élément de cette matrice contient les 4 valeurs des vols symétriques dans le plan transversale : ϕ, $90-\phi$, $90+\phi$, $180-\phi$ et l'élément voisin contient les valeurs des autres 4 voxels symétriques par réflexion (section 7.4.3 et figure 7.2). La matrice gradient *Gr* a été réarrangée et déclarée comme *3D CUDA ARRAY* de type *float8* et dont chaque élément contient les valeurs calculées du gradient pour les 8 voxels symétriques. La matrice des projections (i.e. sinogrammes) mesurées a été réorganisée en p sous-matrices dont chacune contient les projections qui concernent un sous-ensemble (figure 9.2) ; p est le nombre de sous-ensembles. Chaque sous-matrice a été a été implémentée dans la mémoire globale selon la même structure utilisée au chapitre 6 pour arranger en mémoire la matrice de normalisation.

9.5.3 Correction de l'image et filtrage gaussien

Le kernel *ImageUpdate* corrige l'image estimée en la multipliant avec la matrice gradient voxel par voxel, puis en divisant voxel par voxel par la matrice de sensibilité. C'est un kernel qui associe chaque fil d'exécution à une voxel et dont l'exécution est très rapide. Le kernel *GaussianFilter*, qui applique un filtre gaussien 3D à l'image actualisée, est implémenté dans l'espace fréquentiel en utilisant la librairie CUFFT de Nvidia.

Algorithm 2 Pseudo-code du kernel ComputeGradient qui détermine la matrice gradient dans le cas de l'implémentation de l'algorithme AW-LOR-OSEM utilisant la MS calculée en temps réel.

```
__global__ ComputeGradient( ...) // Threads are
associated to basic LORs of subset.
{
 float Prj_Value[N+1]=0.0f //N is the number of symmetries used.
//Forward-projection of the reconstruction image
for i = 1 to number of sub-ray-tracing do
   Calculate Start_point and End_point
   for Voxel j = Start_point to End_point do
      determine Intersection Voxel V_j and RPL_j
      for m = 0 to number of symmetries do
         Vsym_j = Symmetry_m(V_j);
         Prj_Value[m]+ = tex3D(texImage, Vsym_j) * RPL_j;
      end for
   end for
end for

//Determination of projection error.
for m = 0 to number of symmetries do
   Prj_Value[m]=device_ProjectionData[LOR_sym]/Prj_Value[m]
end for
//Back projection of W on device Gradient matrix.
for i = 1 to number of sub-ray-tracing do
   for Voxel j = Start_point to End_point do
      determine Intersection Voxel V_j and RPL_j
      for m = 0 to number of symmetries do
         Vsym_j = Symmetry_m(V_j)
         atomicAdd(device_Gradient[Vsym_j], RPL_j*Prj_Value[m])
      end for
   end for
end for
}
```

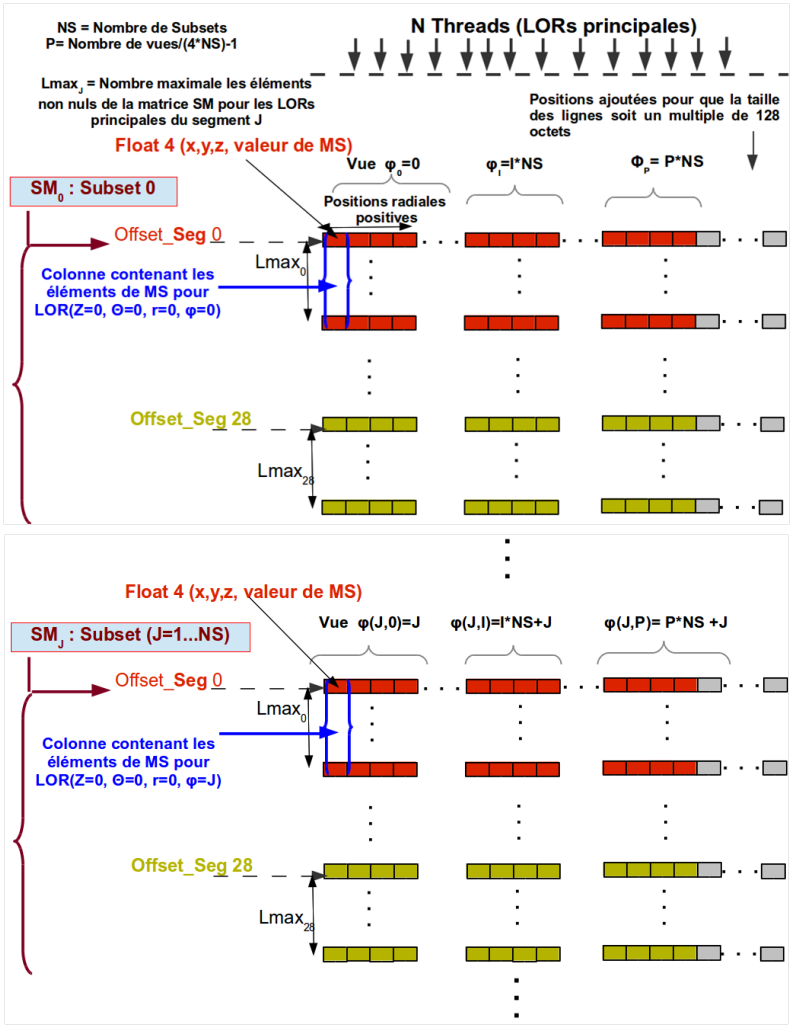


Figure 9.3 – Structure utilisée pour implémenter la matrice MS précalculée.

9.5.4 Stockage de la matrice système dans la mémoire gloabl de GPU

Le défi de l'implantation 2 de l'algorithme AW-LOR-OSEM utilisant une MS précalculée est la réduction de la taille de cette dernière pour être stockée dans la mémoire globale de GPU. En effet, en plus de contenir cette gigantesque matrice (tableau 9.I) durant l'exécution de cet algorithme, la mémoire globale doit contenir la matrice des projections (342 Mo), 11 matrices de sensibilités (11 x 8.06 Mo) pour 11 sous-itérations, la matrice gradientSM (8.06 Mo) et la matrice image (8.06 Mo). De plus, il faut aussi s'assurer que l'accès à ces différentes matrices respecte les conditions de coalescence.

Nous avons expliqué dans la section 2 la stratégie utilisée pour assurer les conditions de coalescence des différentes matrices autres que la MS. Cette section sera donc consacrée pour présenter la méthodologie utilisée afin de réduire la taille de la MS et d'assurer les conditions de coalescence lors de la lecture des données de cette matrice.

La figure 9.3 présente la structure utilisée pour stocker la MS. Pour chaque $LOR(Z=0,\theta,0\leq r,\phi,\leq pi/4)$ principale, la ligne de la MS P_i correspondante est codée sur un vecteur type *float4* qui contient les coefficients non nuls p_{ij} de P_i et les coordonnées (x,y,z) des voxels j concernés. Pour assurer les conditions de coalescence, ces vecteurs sont transformés en colonnes et réorganisés en p sous-matrices dont chacune contient les lignes de la MS qui concernent un sous-ensemble. La matrice globale formée est stockée dans la mémoire de GPU sous forme d'une mémoire linéaire (figure 9.3). Aussi, tous les vecteurs de chaque segment θ ont été stockés dans des colonnes ayant la même longueur $Lmax_\theta$. Cette longueur est celle du vecteur MS obtenu pour la position radiale $r=0$ et elle est égale à la longueur maximale des vecteurs MS du segment θ concerné.

Quoique cette cette stratégie ait l'inconvénient de faire augmenter de l'ordre de 23% la taille de stockage de la MS sur la mémoire globale GPU, elle permet d'une part d'assurer les conditions de coalescence pour la lecture de la MS, et d'autre part de déterminer facilement à partir du numéro de la fil d'exécution l'adresse mémoire du vecteur MS qui correspond au LOR associée au fil concernée. En effet, cette approche permet d'éliminer le besoin de faire transférer à la mémoire globale une matrice de taille de l'ordre de 1 Mo qui doit contenir les adresses mémoires et les longueurs des différents vecteurs de MS. Par conséquent, elle permet d'éviter l'augmentation le temps d'accès à la mémoire globale de GPU qui sera causée par le non respect des conditions de coalescence pour accéder à la MS et par la lecture des adresses dans la mémoire GPU.

9.5.5 Simulations des données

Pour valider et comparer les performances des deux implantations de l'algorithme AW-LOR-OSEM, les données de projections ont été simulées par GATE [90] pour le fantôme décrit à la section 7.4.5.3. Puisque ces deux implantations ne permettent pas pour le moment de corriger pour le diffusé et pour les événements fortuits, les données simulées ont été précorrigées pour ces événements et le nombre de vraies coïncidences générées est 112 millions. Ces données sont stockées dans la matrice de projections sans aucune compression (*span=1,mashing=1*) et la reconstruction a été réalisée pour 11 sous-ensembles OSEM. Les paramètres d'évaluation utilisés

sont le pourcentage de récupération de contraste (CRC) et le rapport signal-bruit (RSB) défini dans la section 7.4.5.3.

9.6 Résultats

9.6.1 Détermination du nombre optimal de rayons par LOR pour l'implantation 1

Pour déterminer le nombre de rayons optimal à sélectionner pour calculer la MS pour l'implantation 1, des matrices de sensibilité pour un fantôme d'eau homogène de 20 cm de diamètre et 20 cm de longueur ont été construites en utilisant différents rayons dans la direction tangentielle (i.e. dans le plan transversal) par LOR et 2 rayons dans la direction axiale. La figure 9.4 présente l'erreur quadratique moyenne relative (RMSE) définie dans la section 8.6.3 entre les matrices de sensibilité obtenues pour différents nombre de rayons dans la direction tangentielle et celle déterminée en utilisant 10 rayons dans la même direction. Nous notons RMSE devient très faible ($< 0.3\%$) et varie peu pour 3 rayons tangentiels et plus. De plus, l'erreur relative $RE(j) = abs(1 - \frac{N_{LOR=3}(j)}{N_{LOR=10}(j)})$ entre les voxels des 2 coupes centrales des matrices de sensibilité obtenues en utilisant 3 et 10 rayons dans la direction tangentielle est au maximum de 1% (figure 9.5). De ce fait, nous avons utilisé 3 rayons tangentiels et de 2 rayons axiaux par LORs pour calculer les coefficients de la MS en temps réel. Le choix de 2 rayons dans la direction axiale est considéré comme optimal puisqu'il permet d'échantillonner sans manquer des voxels de la matrice de reconstruction dont la taille des voxels dans cette direction est $3,15$ mm.

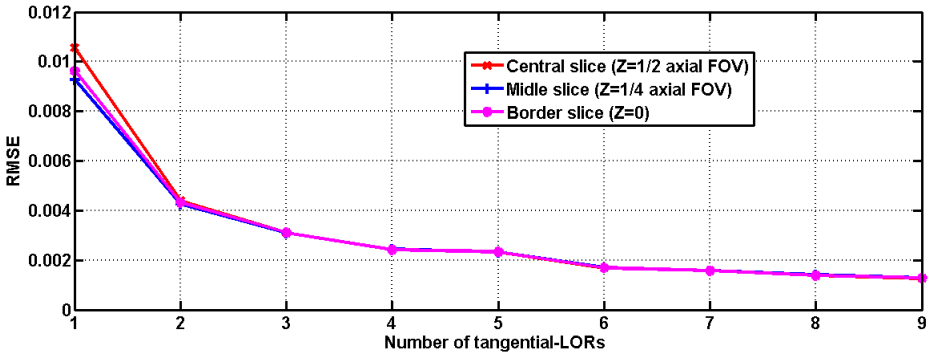
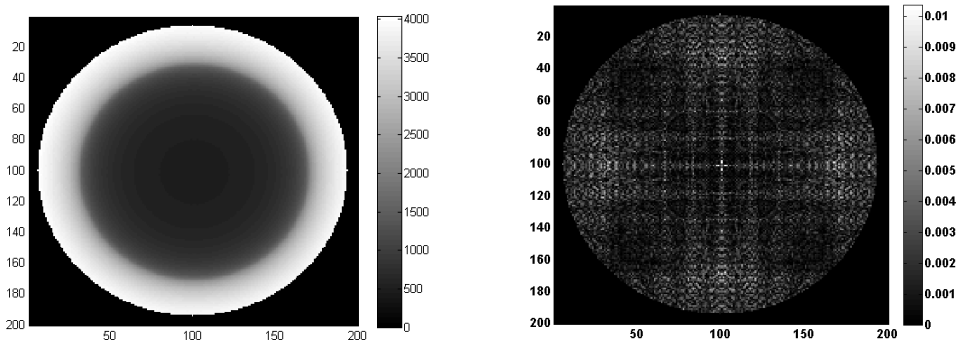


Figure 9.4 – RMSE entre les matrices de sensibilité obtenues pour différents rayons dans la direction tangentielle et celle déterminée en utilisant 10 rayons dans la direction tangentielle.



(a) (b)

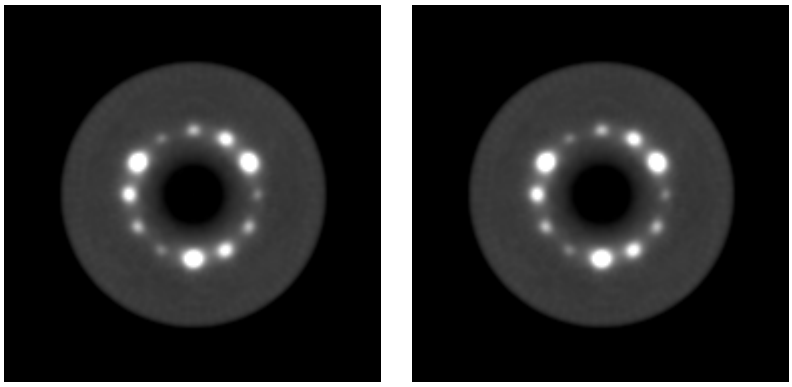
Figure 9.5 – a) La coupe centrale de la matrice de sensibilité du fantôme homogène d'eau calculée en utilisant 3 rayons dans la direction tangentielle et 2 rayons dans la direction axiale par LOR). b) L'erreur relative entre les coupes centrales des matrices de sensibilité obtenues en utilisant 3 et 10 rayons dans la direction tangentielle.

9.6.2 Validation de l'implantation

La Figure 9.6 permet de valider les deux implantations de l'algorithme AW-LOR-OSEM. En effet, les deux images 9.6 a) et b) reconstruites respectivement par l'implantation 1 de l'algorithme ANW-LOR-OSEM qui calcule la MS en temps réel et par l'implantation 2 qui utilise une MS pré-calculée permettent de visualiser correctement les différentes lésions simulées.

9.6.3 Temps de calcul

Le tableau 9.II présente les meilleurs temps de calcul obtenus pour estimer les matrices de sensibilité et pour exécuter les 2 implantations de l'algorithme AW-LOR-OSEM pour 10 sous-ensembles et une itération (i.e. 11 sous-itérations), 112 millions de vrais événements et une matrice de construction de taille 188 x 188 x 57 voxels. Pour l'implanatation 1 qui calcule la MS en temps réel, le meilleur temps est obtenu en utilisant seulement les 8 symétries dans le plan transversale. Ces résultats concordent avec ceux qui sont obtenus au chapitre 6 (voir section Table 7.I). En effet, l'utilisation des 29 symétries axiales fait augmenter le nombre de registres nécessaires par fil d'exécution et, par conséquent, fait diminuer le taux d'occupation des multiprocesseurs. Par contre, pour l'implantation 2 de l'algorithme AW-LOR-OSEM qui pré-calcule MS, il fallait utiliser les 8 symétries transversales et les 29 symétries axiales pour réduire la taille de cette matrice. Donc, nous n'avons pas pu étudier pour cette implantation l'impact sur le temps



a) b)

Figure 9.6 – Images reconstruites par AW-LOR-OSEM : a) la coupe centrale estimée par l'implantation 1 qui détermine MS en temps réel, et b) la coupe à 4.5 cm de la position axiale centrale calculée par l'implantation 2 qui utilise une MS pré-calculée. Les deux images sont estimées en effectuant 2 itérations, 11 sous-ensembles, 112 millions d'événements et sans fonction *atomicAdd()*.

de calcul du nombre des symétries utilisées. Mais nous jugeons que le meilleur temps de calcul sera obtenu en exploitant toutes les symétries pour diminuer fortement la taille de la MS et réduire donc le temps d'accès à la mémoire globale.

Nous notons aussi que le temps de calcul obtenu lorsque la fonction *atomicAdd()* est utilisée, pour éviter la perte des événements, est nettement supérieur à celui enregistré si cette fonction n'est pas appelée. Ce temps est à peu près 5 fois supérieur dans le cas de l'implantation 1, et 2 fois supérieur dans le cas l'implantation 2.

Par ailleurs, la figure 9.7 montre que l'erreur relative entre l'image reconstruite en utilisant la fonction *atomicAdd()* et celle obtenue sans faire appel à cette dernière fonction est inférieure à 0.01%. Par conséquent, nous concluons qu'il n'est pas judicieux d'utiliser cette fonction puisqu'elle engendre une grande perte des performances de calcul.

D'autre part, le temps de calcul pour l'implantation 2 est presque 2 fois celui de l'implantation 1 lorsque la fonction *atomicAdd()* n'est pas utilisée. Ceci s'explique par le fait d'utiliser plus de symétries dans l'implantation 2 pour réduire la taille de la MS cause une diminution du taux d'occupation des multiprocesseurs et crée un étranglement lors de l'accès aux différentes matrices dans la mémoire globale.

Finalement, en comparant les résultats de ce dernier travail avec ceux qui sont obtenus à la section 7.5 du chapitre 7, nous notons que non seulement la reconstruction à partir des sino-

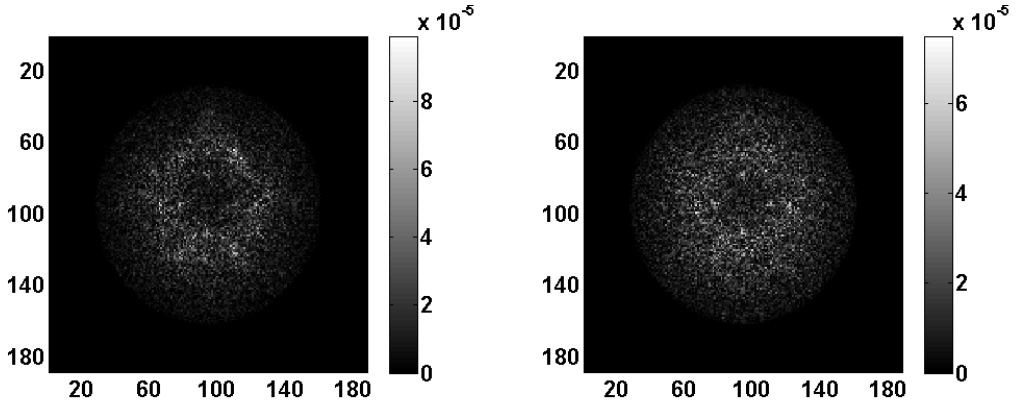
Tableau 9.II – Les temps de calcul des 2 implantations de l'algorithme AW-LOR-OSEM pour 11 sous-ensemble OSEM et 112 millions d'événements.

	Avec utilisation d'*atomicAdd*()	**sans utilisation d'*atomicAdd*()**
Matrices de sensibilité		
MS calculée en temps réel en utilisant les 8 symétries transversales	53 s	10 s
MS pré-calculée en utilisant les 8 symétries transversales et les 29 symétries axiales	42 s	22 s
ANW-LOR-OSEM algorithm, 1 itération		
MS calculée en temps réel en utilisant les 8 symétries transversales	40 s	8 s
MS pré-calculée en utilisant les 8 symétries transversales et les 29 symétries axiales	35 s	18 s

grammes est au moins 8 fois plus rapide que la reconstruction à partir du mode liste, mais aussi l'erreur due à la non-utilisation de la fonction *atomicAdd()* est au moins 100 fois inférieure pour la reconstruction à partir des sinogrammes qu'à partir du mode liste. Cette dernière constatation s'explique par le fait que l'accès à la matrice gradient *Gr* se fait d'une manière aléatoire en mode liste alors qu'elle est ordonnée en mode sinogramme.

9.7 Conclusion et discussion

Dans ce travail, nous avons implanté sur GPU Tesla C2050 avec succès les deux versions de l'algorithme AW-LOR-OSEM. L'implantation 1 qui détermine les coefficients de MS en temps réel durant la reconstruction et l'implantation 2 qui utilise une MS pré-calculée. La MS était déterminée dans les deux cas par la méthode multi-trajectoire de Siddon. Ce choix était dicté dans le cas de l'implantation 2 par la limitation de la mémoire de GPU de Tesla C2050 qui est de 3 GB et qui est insuffisante pour stocker une MS plus précise mais plus grande et qui est calculée par des méthodes plus complexes comme Monte Carlo [253]. Le défi pour les deux implantations est de développer des stratégies pour maximiser le débit de transfert entre la mémoire globale et les multiprocesseurs et minimiser l'échange de données entre ces deux composantes. Nous avons ainsi utilisé les mêmes techniques développées aux chapitres 7 et 8 et qui consistent surtout à :



a) b)

Figure 9.7 – a) Image de l'erreur relative entre les coupes obtenues avec et sans utilisation de la fonction *atomicAdd()* au niveau de l'implantation 1, et b) image de l'erreur relative entre les coupes obtenues avec et sans utilisation de la fonction *atomicAdd()* au niveau de l'implantation 2.

1) utiliser les symétries et faire associer chaque fil d'exécution à une LOR principale et ensuite faire effectuer par le même fil la projection et la rétroprojection pour les LORs associées par symétries. Les performances sont donc améliorées en éliminant l'accès à la mémoire globale pour stocker les résultats intermédiaires de projection pour tous les LORs, 2) réordonner la matrice des données d'acquisition pour assurer les conditions de coalescence d'accès à la mémoire GPU, 3) associer la matrice image à une texture et la réordonner pour que les voxels en relation par symétrie soient voisins et 4) déclarer la matrice gradient comme *3D CUDA ARRAY* de type *float8* pour écrire simultanément les voxels liées par les symétries transversales. Pour l'implantation 2, nous avons réduit la taille de la MS en ne stockant que les coefficients non nuls et nous l'avons réorganisée pour les conditions de coalescence d'accès.

Les temps d'exécution obtenus permettent l'utilisation en clinique de routine l'algorithme AW-LOR-OSEM. Par ailleurs, les performances en temps de calcul obtenus pour l'implantation 1 qui détermine la MS en temps réel sont le double de celle qui utilise une MS pré-calculée. Mais malgré ces résultats, le temps d'exécution de l'implantation 2 est très rapide pour permettre de privilégier son utilisation en clinique puisqu'elle est supérieure en terme de précision de quantification [4, 111, 181, 207, 253]. De plus, les dispositifs GPU modernes tels que Tesla K20X GPU et Tesla K40X sont équipés de mémoire très étendue (respectivement 6 et 12 GB) permettant l'utilisation d'une MS très grande qui sera pré-calculée avec plus de précision en utilisant des

méthodes plus complexes. Le futur travail est de : 1) pré-calculer la MS par la méthode Monte Carlo surtout qu'on dispose à de la puissance de calcule nécessaire pour le faire et des logiciels de modélisation TEP tels que GATE et PET-EGS , et 2) intégrer les fonctions de correction du diffusé et des événements fortuits à notre implantation et la faire exécuter sur les nouveaux dispositifs GPU qui offrent une capacité de mémoire capable de stocker une MS déterminée par Monte Carlo.

CHAPITRE 10

CONCLUSION

La TEP est surtout utilisée en cancérologie pour déterminer le degré de malignité des tumeurs, évaluer leur progression et leur réponse aux traitements et pour détecter des métastases et des tumeurs récidivantes. Pour améliorer la précision de la quantification, les systèmes TEP modernes utilisent un grand nombre de détecteurs solides très performants de petite taille et des systèmes électroniques rapides permettant des acquisitions 3D. Par conséquent, la taille des fichiers des données d'acquisition générées par ces systèmes est énorme, et le temps de reconstruction est devenu long par rapport aux besoins d'utilisation dans un milieu clinique.

Pour réduire ce temps de calcul, les manufacturiers compressent les données d'acquisitions et utilisent des algorithmes de reconstruction dont la convergence est plus rapide, mais dont la quantification est biaisée. Par ailleurs, le diagnostic se base essentiellement sur la mesure du paramètre semi-quantitatif SUV qui quantifie la distribution 3D du traceur alors que ces systèmes ont le potentiel de déterminer *in vivo* avec précision des paramètres physiologiques en faisant la reconstruction 4D. Ces systèmes modernes d'acquisition ont un potentiel d'améliorer le diagnostic en cancérologie et d'élargir les applications cliniques de cette modalité, mais il n'a pas été exploité à cause des contraintes du temps de reconstruction.

10.1 Travaux et résultats

10.1.1 Accélération sur GPU du calcul de la matrice de sensibité pour la reconstruction à partir du mode liste

Dans ce projet, nous nous sommes joints à l'effort de plusieurs chercheurs pour améliorer la quantification en TEP en accélérant la reconstruction sur GPU des algorithmes de reconstruction plus performants mais dont le temps d'exécution est long. Nous avons implémenté sur GPU Tesla C2050 avec succès l'algorithme de reconstruction LM-OSEM incluant le calcul de la matrice de sensibilité qui corrige pour l'atténuation et pour la normalisation. Notre effort a été concentré surtout sur la minimisation du temps de reconstruction de la matrice de sensibilité. En effet, quoique le calcul de cette matrice soit intense dans le cas la reconstruction à partir du mode liste, l'accélération sur les GPUs de sa reconstruction a été négligée par les travaux antérieurs portant

sur l'accélération de l'algorithme LM-OSEM sur GPUs. Nous avons effectué l'implantation pour le système de Philips Gemini GXL qui a 85 millions de LORs en utilisant 6 rayons par LOR afin d'améliorer l'échantillonnage.

Pour atteindre notre objectif, nous avons apporté trois innovations majeures qui sont : 1) diminuer l'accès à la mémoire globale en calculant avec le même *thread* la projection et la rétro-projection pour tous les LORs symétriques ; 2) diminuer la latence de l'accès aléatoire de lecture de la matrice d'atténuation et celui d'écriture dans la matrice de sensibilité en transformant ces deux matrices pour que les voxels symétriques deviennent voisins et 3) enregistrer les données de normalisation (sinogrammes) dans la mémoire globale de manière à ce que les conditions de la coalescence soient respectées. Nous avons aussi démontré que le fait de ne pas utiliser la fonction *atomicAdd()* a un impact minime sur la précision de la quantification, alors que son utilisation pénalise énormément l'efficacité d'exécution en multipliant le temps de calcul au minimum d'un facteur 6.

Le temps que nous avons obtenu pour calculer la matrice de sensibilité de définition de 188 x 188 x 57 voxels est de 9 secondes et il est de 1.1 secondes par million d'événements pour exécuter l'algorithme LM-OSEM. Ce temps permettra d'introduire en clinique quotidienne l'algorithme 3D mode liste LM-OSEM. Il permettra aussi d'envisager la reconstruction dynamique en temps réel et la reconstruction 4D pour le mode liste. Par ailleurs, nous avons noté aussi que le temps par événement pour exécuter LM-OSEM est 9 fois supérieur à celui par LOR pour calculer la matrice de sensibilité.

10.1.2 Développement et accélération sur GPU de l'algorithme Multigrid Multiframe LM-EM

Le travail précédent nous a permis de noter que contrairement au mode sinogramme, l'accélération sur GPU des algorithmes de reconstruction à partir des données en mode liste est très limitée du fait que les symétries ne peuvent pas être utilisées et que l'accès à la mémoire ne peut pas être optimisé pour ce mode. Donc pour surmonter ces obstacles qui entravent l'accélération sur GPU de la reconstruction à partir du mode liste, il faut trouver une solution au niveau de l'algorithme.

Ainsi, nous avons développé et implémenté sur GPU l'algorithme *Multigrid Multiframe* LM-EM (MGMF-LMEM) qui démarre la reconstruction durant l'intervalle de l'acquisition afin d'accélérer l'algorithme LM-EM qui est un algorithme convergent et robuste. En effet, la recon-

struction démarre avec une matrice de faible définition (grands voxels) et lorsque les statistiques d'acquisition s'améliorent après un certain temps d'acquisition, on affine la définition de la matrice de reconstruction en utilisant l'interpolation bilinéaire ou cubique et on répète le processus de reconstruction. Nous avons aussi introduit dans cette implémentation un nouveau critère de convergence dont le temps de détermination sur GPU est négligeable par rapport au temps d'exécution d'une itération de l'algorithme. L'impact sur la précision de la quantification et sur le temps de calcul des différentes méthodes d'interpolation telles que la méthode du voisin le plus proche, l'interpolation bilinéaire et la méthode de l'interpolation cubique B-spline, a été aussi étudié.

Les résultats obtenus montrent que MGMF-LMEM converge vers la même solution que l'algorithme LM-EM avec une vitesse qui est au moins 3 fois plus rapide. Il montre aussi qu'une seule itération de cet algorithme permet d'obtenir un recouvrement de contraste supérieur à 75% du contraste maximal et d'avoir la plus petite erreur des moindres carrés moyenne entre l'image estimée et l'image réelle. Et puisque le temps exécution d'une itération d'un million d'événements est 1.1 seconde pour une matrice de définition de 188 x 188 x 57 voxels, alors l'algorithme MGMF-LMEM que nous avons proposé peut être utilisé comme un algorithme d'un seul passage pour faire de la reconstruction en temps quasi réel pour les acquisitions dynamiques.

10.1.3 Implantation sur GPU de l'algorithme haute résolution 3D AW-LOR-OSEM

La reconstruction à partir du mode liste est mal adaptée aux longues acquisitions pour lesquelles le nombre d'événements est largement supérieur aux nombres des LORs. L'accélération sur GPU de ce mode reste limité par rapport à la reconstruction à partir des sinogrammes. Ainsi, nous avons étudié et exploré la possibilité d'accélérer sur GPU l'algorithme 3D OSEM en utilisant une MS précalculée pour améliorer la quantification. Nous avons donc implémenté sur GPU l'algorithme 3D AW-LOR-OSEM qui est la version haute résolution de l'algorithme OSEM effectuant la reconstruction à partir des sinogrammes sans compression qui intègre la correction de l'atténuation et de la normalisation dans la MS pour préserver la nature stochastique des données d'acquisition.

L'implantation a été effectuée pour les deux méthodes qui consistent à utiliser une MS calculée en temps réel durant la reconstruction et une MS précalculée et stockée dans la mémoire globale dans l'objectif de comparer l'efficacité de calcul entre les deux méthodes. La MS a été calculée par la méthode de multi-trajectoires. Sa taille est de 1.2 Go après compression en utili-

sant les symétries et en ne stockant que les coefficients non nuls. Beaucoup d'énergie a été investi dans l'implantation de la deuxième version utilisant la MS précalculée pour diminuer la latence d'accès à la mémoire globale.

Le temps de calcul pour une itération est de l'ordre de 8 secondes pour l'implantation utilisant une MS calculée en temps réel durant la reconstruction pour 3 rayons dans la direction tangentielle et 2 rayons dans la direction axiale par LOR. Il est de l'ordre de 18 secondes pour l'implantation utilisant une MS précalculée. Nous notons donc que l'efficacité de calcul de la première implémentation est supérieure au double de la deuxième implantation. Cependant, le temps de calcul est très court pour les deux implantations. En effet, pour 11 itérations qui est la valeur typique utilisée en clinique, le temps de calcul est de l'ordre de 2 minutes. Nous pourrons conclure donc que l'utilisation des GPUs permettrait d'utiliser en clinique l'algorithme haute résolution 3D AW-LOR-OSEM se basant sur une MS précalculée pour améliorer la précision de la quantification.

10.2 Travaux futurs

Comme continuation à ce projet, nous suggérons de :

- Implémenter sur GPU les fonctions de correction des événements diffusés et ceux qui sont fortuits. Pour le mode liste, ces fonctions sont complexes et nécessitent un calcul intense. L'implémentation de ces derniers doit être idéalement faite en utilisant un deuxième GPU pour éviter d'augmenter le temps de calcul de la reconstruction. L'estimation du diffusée peut être réalisée en utilisant le code de Monte Carlo sur GPU développé par notre groupe [75].
- Déterminer la matrice des fonctions de dispersion ponctuelle des détecteurs (PSFs : point spread functions) et les intégrer à la reconstruction pour améliorer la récupération du contraste [227].
- Calculer la MS par Monte Carlo pour l'algorithme 3D AW-LOR-OSEM utilisant une MS précalculée pour améliorer la précision de la quantification [253]. Cette matrice serait plus dense que celle calculée par multi-trajectoires et donc sa taille de stockage serait plus grande. Il faut donc utiliser les nouveaux GPUs comme Tesla K20X et Tesla K40 qui disposent d'une DRAM de capacité supérieure à 6 Go.
- Valider les différentes implémentations en utilisant les données cliniques.

– Développer une plateforme graphique qui permet d'intégrer les différents algorithmes et fonctions de reconstruction sur GPU développés dans ce projet.

Nous considérons ce projet comme un travail préliminaire pour la reconstruction 4D qui permet de quantifier avec précision les paramètres physiologiques et dont le temps de calcul est un des obstacles à son utilisation en clinique. Donc, nous suggérons, qu'après avoir maîtrisé la reconstruction 3D sur GPU, de travailler sur l'implémentation sur GPU de la reconstruction 4D dont l'utilisation en clinique permettrait de mieux exploiter le caractère fonctionnel de la TEP. Nous croyons que ce travail doit être fait en collaboration avec les autres groupes qui travaillent sur l'accélération de la reconstruction TEP, surtout le groupe de STIR qui a commencé l'intégration dans son logiciel des *kernels* GPU [219].

BIBLIOGRAPHIE

[1] *Professional Cuda C Programming, author=Cheng, John and Grossman, Max and McKercher, Ty, year=2014, publisher=John Wiley & Sons*.

[2] Michael C Adams, Timothy G Turkington, Joshua M Wilson et Terence Z Wong. A systematic review of the factors affecting accuracy of SUV measurements. *American Journal of Roentgenology*, 195(2):310–320, 2010.

[3] S Agostinelli, J Allison, K ea Amako, J Apostolakis, H Araujo, P Arce, M Asai, D Axen, S Banerjee, G Barrand et al. GEANT4—a simulation toolkit. *Nuclear instruments and methods in physics research section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.

[4] Pablo Aguiar, Magdalena Rafecas, Juan Enrique Ortuño, George Kontaxakis, Andrés Santos, Javier Pavía et Domènec Ros. Geometrical and Monte Carlo projectors in 3D PET reconstruction. *Medical physics*, 37:5691, 2010.

[5] Adam M Alessio et Paul E Kinahan. Application of a spatially variant system model for 3-D whole-body PET image reconstruction. Dans *Biomedical Imaging : From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1315–1318. IEEE, 2008.

[6] Adam M Alessio, Charles W Stearns, Shan Tong, Steven G Ross, Steve Kohlmyer, Alex Ganin et Paul E Kinahan. Application and evaluation of a measured spatially variant system model for PET image reconstruction. *Medical Imaging, IEEE Transactions on*, 29 (3):938–949, 2010.

[7] John Allison, K Amako, J Apostolakis, HAAH Araujo, P Arce Dubois, MAAM Asai, GABG Barrand, RACR Capra, SACS Chauvie, RACR Chytracek et al. Geant4 developments and applications. *Nuclear Science, IEEE Transactions on*, 53(1):270–278, 2006.

[8] Carl D Anderson. Energies of cosmic-ray particles. *Physical Review*, 41(4):405, 1932.

[9] A Andreyev, A Sitek et A Celler. Acceleration of blob-based iterative reconstruction algorithm using Tesla GPU. Dans *Nuclear Science Symposium Conference Record (NSS/MIC), 2009 IEEE*, pages 4095–4098. IEEE, 2009.

[10] Babak A Ardekani, Michael Braun, Brian F Hutton, Iwao Kanno et Hidehiro Iida. Minimum cross-entropy reconstruction of PET images using prior anatomical information. *Physics in Medicine and Biology*, 41(11):2497, 1996.

[11] Jon Petter Asen, Jo Inge Buskenes, CC Nilsen, Andreas Austeng et Sverre Holm. Implementing capon beamforming on a GPU for real-time cardiac ultrasound imaging. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 61(1):76–85, 2014.

[12] Karine Assie, Vincent Breton, Irene Buvat, Claude Comtat, Sebastien Jan, Magalie Krieguer, Delphine Lazaro, Christian Morel, Martin Rey, Giovanni Santin et al. Monte Carlo simulation in PET and SPECT instrumentation using GATE. *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 527(1):180–189, 2004.

[13] Awen AUTRET, Julien Bert, Olivier STRAUSS, Dimitris VISVIKIS et al. Fully 3D PET List-Mode reconstruction including an accurate detector modeling on GPU architecture. Dans *FULLY 3D 2013 : International meeting on fully three dimensional image reconstruction in radiology and nuclear medecine*, 2013.

[14] Awen AUTRET, Julien Bert, BAHI Zakaria, Olivier STRAUSS, Dimitris VISVIKIS et al. Accurate fully 3D list-mode PET reconstruction on multi-GPUs. Dans *RITS 2013 : colloque Recherche en imagerie et technologies pour la santé*, 2013.

[15] Mohammad Reza Ay, Mojtaba Shamsaie Zafarghandi, George Loudos et al. Performance comparison of four commercial GE discovery PET/CT scanners : A monte carlo study using GATE. *Iranian Journal of Nuclear Medicine*, 17(2), 2009.

[16] RD Badawi et PK Marsden. Developments in component-based normalization for 3D PET. *Physics in medicine and biology*, 44(2):571, 1999.

[17] Kristof Baete, Johan Nuyts, Wim Van Paesschen, Paul Suetens et Patrick Dupont. Anatomical-based FDG-PET reconstruction for the detection of hypo-metabolic regions in epilepsy. *Medical Imaging, IEEE Transactions on*, 23(4):510–519, 2004.

[18] B Bai, Q Li, CH Holdsworth, E Asma, YC Tai, A Chatziioannou et RM Leahy. Model-based normalization for iterative 3D PET image reconstruction. *Physics in medicine and biology*, 47(15):2773, 2002.

[19] Dale L Bailey, David W Townsend, Peter E Valk et Michael N Maisey. *Positron emission tomography : basic sciences*. Springer, 2005.

[20] Harrison H. Barrett, T. White et L. Parra. List-mode likelihood. *JOSA A*, 14(11):2914–2923, 1997. URL `http://www.opticsinfobase.org/josaa/abstract.cfm?id=2032`.

[21] Tobias Beisel, Stefan Lietsch et Kris Thielemans. A method for OSEM PET reconstruction on parallel architectures using STIR. Dans *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, pages 4161–4168. IEEE, 2008.

[22] Marcel Beister, Daniel Kolditz et Willi A Kalender. Iterative reconstruction methods in X-ray CT. *Physica medica*, 28(2):94–108, 2012.

[23] Ralf Bergmann, Jens Pietzsch, Frank Fuechtner, Beate Pawelke, Bettina Beuthien-Baumann, Bernd Johannsen et Joerg Kotzerke. 3-O-methyl-6-18F-fluoro-L-dopa, a new tumor imaging agent : investigation of transport mechanism in vitro. *Journal of Nuclear Medicine*, 45(12):2116–2122, 2004.

[24] Julien Bert et Dimitris Visvikis. A fast CPU/GPU ray projector for fully 3D list-mode PET reconstruction. Dans *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*, pages 4126–4130. IEEE, 2011.

[25] Nicolai Bissantz, Bernard A Mair et Axel Munk. A multi-scale stopping criterion for MLEM reconstructions in PET. Dans *Nuclear Science Symposium Conference Record, 2006. IEEE*, volume 6, pages 3376–3379. IEEE, 2006.

[26] Ronald Boellaard. Standards for PET image acquisition and quantitative data analysis. *Journal of nuclear medicine*, 50(Suppl 1):11S–20S, 2009.

[27] Ronald Boellaard, Mike J O'Doherty, Wolfgang A Weber, Felix M Mottaghy, Markus N Lonsdale, Sigrid G Stroobants, Wim JG Oyen, Joerg Kotzerke, Otto S Hoekstra, Jan Pruim et al. FDG PET and PET/CT : EANM procedure guidelines for tumour PET imaging : version 1.0. *European journal of nuclear medicine and molecular imaging*, 37(1):181–200, 2010.

[28] Charles A Bouman et Ken Sauer. A unified approach to statistical tomography using coordinate descent optimization. *Image Processing, IEEE Transactions on*, 5(3):480–492, 1996.

[29] Marco Brambilla, Chiara Secco, Marco Dominietto, Roberta Matheoud, Gianmauro Sacchetti et Eugenio Inglese. Performance characteristics obtained for a new 3-dimensional lutetium oxyorthosilicate–based whole-body PET/CT scanner with the National Electrical Manufacturers Association NU 2-2001 Standard. *Journal of Nuclear Medicine*, 46(12): 2083–2091, 2005.

[30] J. Browne et A.R. De Pierro. A row-action alternative to the EM algorithm for maximizing likelihood in emission tomography. *Medical Imaging, IEEE Transactions on*, 15(5):687–699, 1996. ISSN 0278-0062.

[31] Jolyon Browne et AB De Pierro. A row-action alternative to the EM algorithm for maximizing likelihood in emission tomography. *Medical Imaging, IEEE Transactions on*, 15 (5):687–699, 1996.

[32] Gordon L Brownell et William H Sweet. Localization of brain tumors with positron emitters. *Nucleonics*, 11(11):40–5, 1953.

[33] Irène Buvat. Les limites du SUV. *Médecine Nucléaire*, 31(4):165–172, 2007.

[34] C C. Morel, L Simon, M Krieguer et M Rey. ClearPET Project : LMF specifications. Rapport technique, École Polytechnique Fédérale de Lausanne, 2005.

[35] R.E. Carson, W.C. Barker et CA Johnson. Design of a motion-compensation OSEM list-mode algorithm for resolution-recovery reconstruction for the HRRT. Dans *Nuclear Science Symposium Conference Record*, volume 5, pages 3281–3285. IEEE, 2004.

[36] R.E. Carson, W.C. Barker, J.S. Liow et C.A. Johnson. Design of a motion-compensation OSEM list-mode algorithm for resolution-recovery reconstruction for the HRRT. Dans *Nuclear Science Symposium Conference Record, 2003 IEEE*, volume 5, pages 3281–3285. IEEE, 2003. URL `http://dx.doi.org/10.1109/NSSMIC.2003.1352597`.

[37] I Castiglioni, O Cremonesi, MC Gilardi, V Bettinardi, G Rizzo, A Savi, E Bellotti et

F Fazio. Scatter correction techniques in 3D PET : a Monte Carlo evaluation. *Nuclear Science, IEEE Transactions on*, 46(6):2053–2058, 1999.

[38] Kyle M Champley, Raymond R Raylman et Paul E Kinahan. Advancements to the planogram frequency–distance rebinning algorithm. *Inverse problems*, 26(4):045008, 2010.

[39] Ji-Ho Chang, John MM Anderson et JT Votaw. Regularized image reconstruction algorithms for positron emission tomography. *Medical Imaging, IEEE Transactions on*, 23(9): 1165–1175, 2004.

[40] Chung-Ming Chen. An efficient four-connected parallel system for PET image reconstruction. *Parallel Computing*, 24(9):1499–1522, 1998.

[41] Simon R Cherry, Magnus Dahlbom et Edward J Hoffman. Evaluation of a 3D reconstruction algorithm for multi-slice PET scanners. *Physics in medicine and biology*, 37(3):779, 1992.

[42] Claude Comtat, Paul E Kinahan, Jeffrey A Fessler, Thomas Beyer, David W Townsend, Michel Defrise et Christian Michel. Clinically feasible reconstruction of 3D whole-body PET/CT data using blurred anatomical labels. *Physics in Medicine and Biology*, 47(1):1, 2002.

[43] Allan Macleod Cormack. Representation of a function by its line integrals, with some radiological applications. *Journal of Applied Physics*, 34(9):2722–2727, 1963.

[44] Jingyu Cui, Guillem Pratx, Bowen Meng et C.S. Levin. Distributed MLEM : An Iterative Tomographic Image Reconstruction Algorithm for Distributed Memory Architectures. *Medical Imaging, IEEE Transactions on*, 32(5):957–967, 2013. ISSN 0278-0062.

[45] Jingyu Cui, Guillem Pratx, Sven Prevrhal et Craig Levin. Fully 3D list-mode time-of-flight PET image reconstruction on GPUs using CUDA. *Medical Physics*, 38(12):6775, 2011.

[46] Margaret E Daube-Witherspoon, Samuel Matej, Matthew E Werner, Suleman Surti et Joel S Karp. Comparison of list-mode and DIRECT approaches for time-of-flight PET reconstruction. *Medical Imaging, IEEE Transactions on*, 31(7):1461–1471, 2012.

[47] Margaret E Daube-Witherspoon et Gerd Muehllehner. Treatment of axial data in three-dimensional PET. *J Nucl Med*, 28(11):1717–1724, 1987.

[48] Jan De Beenhouwer, Steven Staelens, Dirk Kruecker, Ludovic Ferrer, Yves D'Asseler, Ignace Lemahieu et Fernando R Rannou. Cluster computing software for GATE simulations. *Medical physics*, 34(6):1926–1933, 2007.

[49] Hugo WAM de Jong, Floris HP van Velden, Reina W Kloet, Fred L Buijs, Ronald Boellaard et Adriaan A Lammertsma. Performance evaluation of the ECAT HRRT : an LSO-LYSO double layer high resolution, high sensitivity scanner. *Physics in medicine and biology*, 52(5):1505, 2007.

[50] Dirk De Ruysscher et Carl-Martin Kirsch. PET scans in radiotherapy planning of lung cancer. *Radiotherapy and Oncology*, 96(3):335–338, 2010.

[51] Michel Defrise, Paul E Kinahan, David W. Townsend, Christian Michel, Merence Sibomana et DF Newport. Exact and approximate rebinning algorithms for 3-D PET data. *Medical Imaging, IEEE Transactions on*, 16(2):145–158, 1997.

[52] Arthur P Dempster, Nan M Laird et Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[53] Paul AM Dirac. Quantised singularities in the electromagnetic field. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 133(821):60–72, 1931.

[54] Anders Eklund, Paul Dufort, Daniel Forsberg et Stephen M LaConte. Medical image processing on the GPU–Past, present and future. *Medical image analysis*, 17(8):1073–1094, 2013.

[55] Hakan Erdogan et Jeffrey A Fessler. Ordered subsets algorithms for transmission tomography. *Physics in medicine and biology*, 44(11):2835, 1999.

[56] Samuel España, JL Herraiz, Esther Vicente, Juan José Vaquero, Manuel Desco et José Manuel Udías. PeneloPET, a Monte Carlo PET simulation tool based on PENELOPE : features and validation. *Physics in medicine and biology*, 54(6):1723, 2009.

[57] T. Felder, M. Blume, J. F. Oliver et M Rafecas. ML-EM Implementation on a GPU : Avoiding Simultaneous Read-Modify-Write Processes. Dans *Proceedings of 10th Fully 3D Meeting and 2nd HPIR Workshop*, pages 65–68, 2009.

[58] Jeffrey A Fessler et Alfred O Hero III. Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms. *Image Processing, IEEE Transactions on*, 4(10):1417–1429, 1995.

[59] Nanette M Freedman, Senthil K Sundaram, Karen Kurdziel, Jorge A Carrasquillo, Millie Whatley, Joann M Carson, David Sellers, Steven K Libutti, James C Yang et Stephen L Bacharach. Comparison of SUV and Patlak slope for monitoring of cancer therapy using serial PET scans. *European journal of nuclear medicine and molecular imaging*, 30(1): 46–53, 2003.

[60] Michaela Gaens, Julien Bert, Uwe Pietrzyk, N Jon Shah et Dimitris Visvikis. GPU-accelerated Monte Carlo based scatter correction in brain PET/MR. Dans *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2013 IEEE*, pages 1–3. IEEE, 2013.

[61] Anastasios Gaitanis, George Kontaxakis, George Spyrou, George Panayiotakis et George Tzanakos. PET image reconstruction : A stopping rule for the MLEM algorithm based on properties of the updating coefficients. *Computerized Medical Imaging and Graphics*, 34 (2):131–141, 2010.

[62] Richard Gordon, Robert Bender et Gabor T Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of theoretical Biology*, 29(3):471–481, 1970.

[63] Erwan Gravier, Yongyi Yang et Mingwu Jin. Tomographic reconstruction of dynamic cardiac image sequences. *Image Processing, IEEE Transactions on*, 16(4):932–942, 2007.

[64] Peter J Green. Bayesian reconstructions from emission tomography data using a modified EM algorithm. *Medical Imaging, IEEE Transactions on*, 9(1):84–93, 1990.

[65] Corinne J Groiselle et Stephen J Glick. 3D PET list-mode iterative reconstruction using time-of-flight information. Dans *Nuclear Science Symposium Conference Record, 2004 IEEE*, volume 4, pages 2633–2638. IEEE, 2004.

[66] N. Grotus, Andrew J Reader, S Stute, JC Rosenwald, P. Giraud et Irene Buvat. Fully 4D list-mode reconstruction applied to respiratory-gated PET scans. *Physics in Medicine and Biology*, 54:1705, 2009. URL `http://iopscience.iop.org/0031-9155/54/6/020`.

[67] Per Christian Hansen et Toke Koldborg Jensen. Noise propagation in regularizing iterations for image deblurring. *Electronic Transactions on Numerical Analysis*, 31:204–220, 2008.

[68] Tom Hebert et Richard Leahy. A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. *Medical Imaging, IEEE Transactions on*, 8(2): 194–202, 1989.

[69] J.L. Herraiz, S. Espaa, S. Garcia, R. Cabido, A.S. Montemayor, M. Desco, J.J. Vaquero et J. M. Udias. GPU acceleration of a fully 3D Iterative Reconstruction Software for PET using CUDA. Dans *Nuclear Science Symposium Conference Record (NSS/MIC), 2009 IEEE*, pages 4064–4067, 2009. ISSN 1095-7863.

[70] JL Herraiz, Samuel España, R Cabido, AS Montemayor, Manuel Desco, Juan José Vaquero et José Manuel Udías. GPU-based fast iterative reconstruction of fully 3-D PET Sinograms. *Nuclear Science, IEEE Transactions on*, 58(5):2257–2263, 2011.

[71] JL Herraiz, Samuel España, Juan José Vaquero, Manuel Desco et José Manuel Udías. FIRST : Fast iterative reconstruction software for PET tomography. *Physics in medicine and biology*, 51(18):4547, 2006.

[72] Ken Herrmann, Katja Ott, Andreas K Buck, Florian Lordick, Dirk Wilhelm, Michael Souvatzoglou, Karen Becker, Tibor Schuster, Hans-Jürgen Wester, Jörg R Siewert et al. Imaging gastric cancer with PET and the radiotracers 18F-FLT and 18F-FDG : a comparative analysis. *Journal of Nuclear Medicine*, 48(12):1945–1950, 2007.

[73] David M Higdon, James E Bowsher, Valen E Johnson, Timothy G Turkington, David R Gilland et Ronald J Jaszczak. Fully Bayesian estimation of Gibbs hyperparameters for emission computed tomography data. *Medical Imaging, IEEE Transactions on*, 16(5): 516–526, 1997.

[74] Sami Hissoiny, Benoît Ozell, Hugo Bouchard et Philippe Després. GPUMCD : A new GPU-oriented Monte Carlo dose calculation platform. *Medical Physics*, 38:754, 2011.

[75] Sami Hissoiny, Benot Ozell, Hugo Bouchard et Philippe Desprs. GPUMCD : A new GPU-oriented Monte Carlo dose calculation platform. *Medical Physics*, 38(2):754–764, 2011. URL `http://link.aip.org/link/?MPH/38/754/1`. arXiv :1101.1245v1.

[76] Edward J Hoffman, Thomas M Guerrero, Guido Germano, WM Digby et M Dahlbom. PET system calibrations and corrections for quantitative and spatially accurate images. *Nuclear Science, IEEE Transactions on*, 36(1):1108–1112, 1989.

[77] IK Hong, ST Chung, HK Kim, YB Kim, YD Son et ZH Cho. Ultra fast symmetry and SIMD-based projection-backprojection (SSP) algorithm for 3-D PET image reconstruction. *Medical Imaging, IEEE Transactions on*, 26(6):789–803, 2007.

[78] GN Hounsfield. A Method of and Apparatus for Examination of a Body by Radiation Such as X-or Gamma-Radiation. *London, not paginated*, 1972.

[79] Yu-Han H Hsu, Gregory Z Ferl et Chee M Ng. GPU-accelerated nonparametric kinetic analysis of DCE-MRI data from glioblastoma patients treated with bevacizumab. *Magnetic resonance imaging*, 31(4):618–623, 2013.

[80] Z. Hu, W. Wang, E. E. Gualtieri, Y. L. Hsieh, Joel S Karp, S. Matej, M.J. Parma, C.H. Tung, E.S. Walsh, M. Werner et D. Gagnon. An LOR-based fully-3D PET image reconstruction using a blob-basis function. Dans *Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE*, volume 6, pages 4415–4418, 2007. ISSN 1095-7863.

[81] Z Hu, W Wang, EE Gualtieri, YL Hsieh, JS Karp, S Matej, MJ Parma, CH Tung, ES Walsh, M Werner et al. An LOR-based fully-3D PET image reconstruction using a blob-basis function. Dans *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, volume 6, pages 4415–4418. IEEE, 2007.

[82] Sung-Cheng Huang. Image oscillation reduction and convergence acceleration for OS-EM reconstruction [PET imaging]. *Nuclear Science, IEEE Transactions on*, 46(3):603–607, 1999. ISSN 0018-9499.

[83] Teng-Yi Huang, Yu-Wei Tang et Shiun-Ying Ju. Accelerating image registration of MRI by GPU-based parallel computation. *Magnetic resonance imaging*, 29(5):712–716, 2011.

[84] H Malcolm Hudson et Richard S Larkin. Accelerated image reconstruction using ordered subsets of projection data. *Medical Imaging, IEEE Transactions on*, 13(4):601–609, 1994.

[85] Ronald H Huesman, Gregory J Klein, William W Moses, Jinyi Qi, Bryan W Reutter et Patrick RG Virador. List-mode maximum-likelihood reconstruction applied to positron emission mammography (PEM) with irregular sampling. *Medical Imaging, IEEE Transactions on*, 19(5):532–537, 2000.

[86] George J Hunter, Leena M Hamberg, Nathaniel M Alpert, Noah C Choi, Alan J Fischman et al. Simplified measurement of deoxyglucose utilization rate. *Journal of Nuclear Medicine*, 37(6):950–954, 1996.

[87] M. Jacobson, R. Levkovitz, A. Ben-Tal, K. Thielemans, T. Spinks, D. Belluzzo, E. Pagani, V. Bettinardi, MC Gilardi, A. Zverovich et al. Enhanced 3D PET OSEM reconstruction using inter-update Metz filtering*. *Physics in Medicine and Biology*, 45:2417, 2000.

[88] Mark S. Jacobson, R. Levkovitz, A. Ben-Tal, K. Thielemans, T. Spinks, D. Belluzzo, E. Pagani, V. Bettinardi, MC Gilardi, A. Zverovich et al. Enhanced 3D PET OSEM reconstruction using inter-update Metz filtering. *Physics in Medicine and Biology*, 45(8):2417–2439, 2000. URL `http://www.iop.org/EJ/article/0031-9155/45/8/325/m00825.pdf?request-id=1d66beb8-1e1f-4883-a510-51d4fc9dec65`.

[89] S Jan, D Benoit, E Becheva, T Carlier, F Cassol, P Descourt, T Frisson, L Grevillot, L Guigues, L Maigne et al. GATE V6 : a major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy. *Physics in medicine and biology*, 56 (4):881, 2011.

[90] S. Jan, G. Santin, D. Strul, S. Staelens, K. Assie, D. Autret, S. Avner, R. Barbier, M. Bardies, PM Bloomfield et al. GATE : a simulation toolkit for PET and SPECT. *Physics in medicine and biology*, 49:4543, 2004. URL `http://stacks.iop.org/0031-9155/49/4543`.

[91] S Jan, G Santin, D Strul, Steven Staelens, K Assie, D Autret, S Avner, R Barbier,

M Bardies, PM Bloomfield et al. GATE : a simulation toolkit for PET and SPECT. *Physics in medicine and biology*, 49(19):4543, 2004.

[92] Toke Koldborg Jensen et Per Christian Hansen. Iterative regularization with minimum-residual methods. *BIT Numerical Mathematics*, 47(1):103–120, 2007.

[93] Xun Jia, Yifei Lou, Ruijiang Li, William Y Song et Steve B Jiang. GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation. *Medical physics*, 37(4):1757–1760, 2010.

[94] S. Kaczmarz. Angen"aherte Auflösung von Systemen linearer Gleichungenn. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A. Sciences Mathématiques*, A3:355–357, 1937.

[95] Dan J Kadrmas. Statistically regulated and adaptive EM reconstruction for emission computed tomography. *Nuclear Science, IEEE Transactions on*, 48(3):790–798, 2001.

[96] Dan J Kadrmas. LOR-OSEM : statistical PET reconstruction from raw line-of-response histograms. *Physics in medicine and biology*, 49(20):4731, 2004.

[97] Dan J Kadrmas. Rotate-and-slant projector for fast LOR-based fully-3-D iterative PET reconstruction. *Medical Imaging, IEEE Transactions on*, 27(8):1071–1083, 2008.

[98] Mustafa E Kamasak, Charles A Bouman, Evan D Morris et Ken Sauer. Direct reconstruction of kinetic parameter images from dynamic PET data. *Medical Imaging, IEEE Transactions on*, 24(5):636–650, 2005.

[99] Vibhu Kapoor, Barry M McCook et Frank S Torok. An Introduction to PET-CT Imaging1. *Radiographics*, 24(2):523–543, 2004.

[100] George A Kastis, Anastasios Gaitanis, Yolanda Fernandez, George Kontaxakis et Athanassios S Fokas. Evaluation of a spline reconstruction technique : Comparison with FBP, MLEM and OSEM. Dans *Nuclear Science Symposium Conference Record (NSS/MIC), 2010 IEEE*, pages 3282–3287. IEEE, 2010.

[101] Kyung Sang Kim et Jong Chul Ye. Fully 3D iterative scatter-corrected OSEM for HRRT PET using a GPU. *Physics in Medicine and Biology*, 56(15):4991, 2011. URL `http://stacks.iop.org/0031-9155/56/i=15/a=021`.

[102] Kyung Sang Kim et Jong Chul Ye. Fully 3D iterative scatter-corrected OSEM for HRRT PET using a GPU. *Physics in medicine and biology*, 56(15):4991, 2011.

[103] Seokhyun Kim, Hak-yeol Sohn, Jin Ho Chang, Tai-kyoung Song et Yangmo Yoo. A PC-based fully-programmable medical ultrasound imaging system using a graphics processing unit. Dans *Ultrasonics Symposium (IUS), 2010 IEEE*, pages 314–317. IEEE, 2010.

[104] Paul E Kinahan et JG Rogers. Analytic 3D image reconstruction using all detected events. *Nuclear Science, IEEE Transactions on*, 36(1):964–968, 1989.

[105] F Lamare, MJ Ledesma Carbayo, T Cresson, G Kontaxakis, A Santos, C Cheze Le Rest, AJ Reader et D Visvikis. List-mode-based reconstruction for respiratory motion correction in PET using non-rigid body transformations. *Physics in medicine and biology*, 52(17): 5187, 2007.

[106] F. Lamare, A. Turzo, Y. Bizais, C Cheze Le Rest et D. Visvikis. Validation of a Monte Carlo simulation of the Philips Allegro/GEMINI PET systems using GATE. *Phys. Med. Biol.*, 51(4):943–962, 2006. URL `http://stacks.iop.org/0031-9155/51/943`.

[107] Kenneth Lange. Convergence of EM image reconstruction algorithms with Gibbs smoothing. *Medical Imaging, IEEE Transactions on*, 9(4):439–446, 1990.

[108] Richard M Leahy et Jinyi Qi. Statistical approaches in quantitative positron emission tomography. *Statistics and Computing*, 10(2):147–165, 2000.

[109] Byeonghun Lee, Ho Lee et Yeong Gil Shin. Fast hybrid CPU-and GPU-based CT reconstruction algorithm using air skipping technique. *Journal of X-ray science and technology*, 18(3):221–234, 2010.

[110] T.M. Lehmann, C. Gonner et K. Spitzer. Survey : Interpolation methods in medical image processing. *Medical Imaging, IEEE Transactions on*, 18(11):1049–1075, 1999. URL `http://dx.doi.org/10.1109/42.816070`.

[111] J-D Leroux, C Thibaudeau, R Lecomte et R Fontaine. Fast, accurate and versatile Monte Carlo method for computing system matrix. Dans *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, volume 5, pages 3644–3648. IEEE, 2007.

[112] R. Levkovilz, D. Falikman, M. Zibulevsky, A. Ben-Tal et A. Nemirovski. The design and implementation of COSEN, an iterative algorithm for fully 3-D listmode data. *Medical Imaging, IEEE Transactions on*, 20(7):633–642, 2002. ISSN 0278-0062.

[113] Thomas K Lewellen. The challenge of detector designs for PET. *American Journal of Roentgenology*, 195(2):301–309, 2010.

[114] Robert M Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physics in Medicine and Biology*, 37(3):705, 1992.

[115] Robert M Lewitt et Samuel Matej. Overview of methods for image reconstruction from projections in emission computed tomography. *Proceedings of the IEEE*, 91(10):1588–1611, 2003.

[116] Robert M Lewitt, Gerd Muehllehner et Joel S Karp. Three-dimensional image reconstruction for PET by multi-slice rebinning and axial image filtering. *Physics in medicine and biology*, 39(3):321, 1994.

[117] Quanzheng Li, Evren Asma, Sangtae Ahn et Richard M Leahy. A fast fully 4-D incremental gradient reconstruction algorithm for list mode PET data. *Medical Imaging, IEEE Transactions on*, 26(1):58–67, 2007.

[118] Tianfang Li, Brian Thorndyke, Eduard Schreibmann, Yong Yang et Lei Xing. Model-based image reconstruction for four-dimensional PET. *Medical physics*, 33:1288, 2006.

[119] Xiang Li, Jun Ni et Ge . Parallel iterative cone beam CT image reconstruction on a PC cluster. *Journal of X-Ray Science and Technology*, 13(2):63–72, 2005.

[120] Xuan Liu, Claude Comtat, C. Michel, Paul E. Kinahan, M. Defrise et DW Townsend. Comparison of 3-D reconstruction with 3D-OSEM and with FORE+OSEM for PET. *Medical Imaging, IEEE Transactions on*, 20(8):804–814, 2001. ISSN 0278-0062.

[121] Jorge Llacer et Eugene Veklerov. Feasible images and practical stopping rules for iterative algorithms in emission tomography. *Medical Imaging, IEEE Transactions on*, 8(2):186–193, 1989.

[122] Jean Logan, Joanna S Fowler, Nora D Volkow, Alfred P Wolf, Stephen L Dewey, David J Schlyer, Robert R MacGregor, Robert Hitzemann, Bernard Bendriem, S John Gatley et al.

Graphical analysis of reversible radioligand binding from time-activity measurements applied to [N-11C-methyl]-(-)-cocaine PET studies in human subjects. *Journal of Cerebral Blood Flow & Metabolism*, 10(5):740–747, 1990.

[123] Michael MacManus, Ursula Nestle, Kenneth E Rosenzweig, Ignasi Carrio, Cristina Messa, Otakar Belohlavek, Massimo Danna, Tomio Inoue, Elizabeth Deniaud-Alexandre, Stefano Schipani et al. Use of PET and PET/CT for radiation therapy planning : IAEA expert report 2006–2007. *Radiotherapy and Oncology*, 91(1):85–94, 2009.

[124] L Maigne et Y Perrot. GATE tutorial, EPIKH Training Course on GATE and Grid Computing in Algeria. 2012.

[125] S. Matej, Suleman Surti, S. Jayanthi, Margaret E Daube-Witherspoon, Robert M Lewitt et Joel S Karp. Efficient 3-D TOF PET Reconstruction Using View-Grouped Histo-Images : DIRECT–Direct Image Reconstruction for TOF. *Medical Imaging, IEEE Transactions on*, 28(5):739–751, 2009.

[126] Samuel Matej et Robert M Lewitt. Efficient 3D grids for image reconstruction using spherically-symmetric volume elements. *Nuclear Science, IEEE Transactions on*, 42(4): 1361–1370, 1995.

[127] Samuel Matej, Suleman Surti, Shridhar Jayanthi, Margaret E Daube-Witherspoon, Robert M Lewitt et Joel S Karp. Efficient 3-D TOF PET Reconstruction Using View-Grouped Histo-Images : DIRECT—Direct Image Reconstruction for TOF. *Medical Imaging, IEEE Transactions on*, 28(5):739–751, 2009.

[128] Julian Matthews, Dale Bailey, Pat Price et Vin Cunningham. The direct calculation of parametric images from dynamic PET data using maximum-likelihood iterative reconstruction. *Physics in medicine and biology*, 42(6):1155, 1997.

[129] Geoffrey McLachlan et Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[130] Andrew McLennan, A. Reilhac et M. Brady. SORTEO : Monte carlo-based simulator with list-mode capabilities. Dans *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3751 –3754, 2009. ISSN 1557-170X.

[131] Steven R Meikle, Brian F Hutton, Dale L Bailey, Patrick K Hooper et Michael J Fulham. Accelerated EM reconstruction in total-body PET : potential for improving tumour detectability. *Physics in medicine and biology*, 39(10):1689, 1994.

[132] Steven R Meikle, Julian C Matthews, Vincent J Cunningham, Dale L Bailey, Lefteris Livieratos, Terry Jones et Pat Price. Parametric image reconstruction using spectral analysis of PET projection data. *Physics in medicine and biology*, 43(3):651, 1998.

[133] Luis Mendes, Nuno Ferreira et Claude Comtat. A multiscale-multiframe approach to 3D PET data reconstruction. Dans *Proceedings of The 11th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine and the 3rd Workshop on High Performance Image Reconstruction*, pages 411–413, 2011.

[134] Luis Mendes, Nuno Ferreira et Claude Comtat. A multiscale-multiframe approach to 3D PET data reconstruction. Dans *Proceedings of The 11th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine and the 3rd Workshop on High Performance Image Reconstruction*, pages 411–413, 2011.

[135] Luis Mendes, Nuno Ferreira et Claude Comtat. An overview on the multiscale/multiframe reconstruction for Positron Emission Tomography. Dans *Bioengineering (ENBENG), 2012 IEEE 2nd Portuguese Meeting in*, pages 1–4, 2012.

[136] C. Michel, X. Liu, S. Sanabria, M. Lonneux, M. Sibomana, A. Bol, C. Comtat, PE Kinahan, DW Townsend et M. Defrise. Weighted schemes applied to 3D-OSEM reconstruction in PET. Dans *Nuclear Science Symposium, 1999. Conference Record*, volume 3, pages 1152–1157. IEEE, 1999.

[137] Brian W Miller, Lars R Furenlid, Stephen K Moore, H Bradford Barber, Vivek V Nagarkar et Harrison H Barrett. System integration of FastSPECT III, a dedicated SPECT rodent-brain imager based on BazookaSPECT detector technology. Dans *Nuclear Science Symposium Conference Record (NSS/MIC), 2009 IEEE*, pages 4004–4008. IEEE, 2009.

[138] Brian W Miller, Roel Van Holen, Harrison H Barrett et Lars R Furenlid. A system calibration and fast iterative reconstruction method for next-generation SPECT imagers. *Nuclear Science, IEEE Transactions on*, 59(5):1990–1996, 2012.

[139] Sascha Moehrs, Michel Defrise, Nicola Belcari, Alberto Del Guerra, Antonietta Bartoli, Serena Fabbri et Gianluigi Zanetti. Multi-ray-based system matrix generation for 3D PET reconstruction. *Physics in medicine and biology*, 53(23):6925, 2008.

[140] Erkan Ü Mumcuoglu, Richard M Leahy et Simon R Cherry. Bayesian reconstruction of PET images : methodology and performance analysis. *Physics in medicine and Biology*, 41(9):1777, 1996.

[141] E.U. Mumcuoglu, Richard M Leahy, Simon R Cherry et Zhenyu Zhou. Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images. *Medical Imaging, IEEE Transactions on*, 13(4):687–701, 1994. ISSN 0278-0062.

[142] Moulay Ali Nassiri, Jean-Franois Carrier et Philippe Després. Fast Reconstruction of High-Resolution Attenuation-Weighted Line-of-Response 3D OSEM PET Images on the GPU. Nuclear Science Symposium and Medical Imaging Conference, 23 October - 29 November 2011. Valencia, Spain.

[143] Moulay Ali Nassiri, Philippe Després, Sami Hissoiny et Jean-Franois Carrier. Fast Computation of High Resolution LOR-Based 3D OSEM PET Algorithm Using the GPU Device. AAPM Annual Meeting, 31 August - 4 July 2011. Vancouver, BC.

[144] Moulay Ali Nassiri, Sami Hissoiny, Jean-Franois Carrier et Philippe Després. Fast GPU-Based Computation of the Sensitivity Matrix for a PET List-Mode OSEM Algorithm. Dans *Proceedinds of the 11th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 11-15 July 2011. Potsdam, Germany.

[145] Moulay Ali Nassiri, Sami Hissoiny, Jean-Franois Carrier et Philippe Després. Fast GPU-Based Computation of the Sensitivity Matrix for a PET List-Mode OSEM Algorithm. *Physics in Medicine and Biology*, 57(19):6279, 2012. URL `http://stacks.iop.org/0031-9155/57/i=19/a=6279`.

[146] Moulay Ali Nassiri, Sami Hissoiny, Jean-François Carrier et Philippe Després. Fast GPU-Based Computation of the Sensitivity Matrix for a PET List-Mode OSEM Algorithm. Dans *Proceedings of The 11th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine and the 3rd Workshop on High Performance Image Reconstruction*, pages 27–30, 2011.

[147] *NEMA NU 2-2007. Performance measurements of positron emission tomographs*. National Electrical Manufacturers Association, 2007.

[148] Van-Giang Nguyen, Jieun Jeong et Soo-Jin Lee. GPU-accelerated iterative 3D CT reconstruction using exact ray-tracing method for both projection and backprojection. Dans *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2013 IEEE*, pages 1–4. IEEE, 2013.

[149] Thomas E Nichols, Jinyi Qi, Evren Asma et Richard M Leahy. Spatiotemporal reconstruction of list-mode PET data. *Medical Imaging, IEEE Transactions on*, 21(4):396–404, 2002.

[150] John Nickolls et William J Dally. The GPU computing ERA. *Micro, IEEE*, 30(2):56–69, 2010.

[151] CUDA Nvidia. CUDA C Programming Guide 5.5. *NVIDIA Corporation, Jul*, 2013.

[152] CUDA Nvidia. Reference manual 5.5. *NVIDIA Corporation, Jul*, 2013.

[153] TR Oakes, V Sossi et TJ Ruth. Normalization for 3D PET with a low-scatter planar source and measured geometric factors. *Physics in medicine and biology*, 43(4):961, 1998.

[154] Seungseok Oh, Charles A Bouman et Kevin J Webb. Multigrid tomographic inversion with variable resolution data and image spaces. *Image Processing, IEEE Transactions on*, 15(9):2805–2819, 2006.

[155] Seungseok Oh, Adam B Milstein, Charles A Bouman et Kevin J Webb. A general framework for nonlinear multigrid inversion. *Image Processing, IEEE Transactions on*, 14(1): 125–140, 2005.

[156] John M Ollinger et Andrew S Goggin. Maximum likelihood reconstruction in fully 3D PET via the SAGE algorithm. Dans *Nuclear Science Symposium, 1996. Conference Record., 1996 IEEE*, volume 3, pages 1594–1598. IEEE, 1996.

[157] Xiaolong Ouyang, Wing H Wong, Valen E Johnson, Xiaoping Hu et Chin-Tu Chen. Incorporation of correlated structural images in PET image reconstruction. *Medical Imaging, IEEE Transactions on*, 13(4):627–640, 1994.

[158] Xiaochuan Pan, Dan Xia, Yu Zou et Lifeng Yu. A unified analysis of FBP-based algorithms in helical cone-beam and circular cone-and fan-beam scans. *Physics in medicine and biology*, 49(18):4349, 2004.

[159] Vladimir Y Panin, Frank Kehren, Christian Michel et Michael Casey. Fully 3-D PET reconstruction with system matrix derived from point source measurements. *Medical Imaging, IEEE Transactions on*, 25(7):907–921, 2006.

[160] HG Park, YG Shin et H Lee. A Fully GPU-Based Ray-Driven Backprojector via a Ray-Culling Scheme with Voxel-Level Parallelization for Cone-Beam CT Reconstruction. *Technology in cancer research & treatment*, 2014.

[161] J.A. Parker, R.V. Kenyon et D.E. Troxel. Comparison of interpolating methods for image resampling. *Medical Imaging, IEEE Transactions on*, 2(1):31–39, 1983. URL `http://dx.doi.org/10.1109/TMI.1983.4307610`.

[162] L. Parra et H.H. Barrett. List-mode likelihood : EM algorithm and image quality estimation demonstrated on 2-D PET. *Medical Imaging, IEEE Transactions on*, 17(2):228–235, 1998.

[163] Clifford S Patlak, Ronald G Blasberg, Joseph D Fenstermacher et al. Graphical evaluation of blood-to-brain transfer constants from multiple-time uptake data. *J Cereb Blood Flow Metab*, 3(1):1–7, 1983.

[164] Arnold C Paulino, Mary Koshy, Rebecca Howell, David Schuster et Lawrence W Davis. Comparison of CT-and FDG-PET-defined gross tumor volume in intensity-modulated radiotherapy for head-and-neck cancer. *International Journal of Radiation Oncology* Biology* Physics*, 61(5):1385–1392, 2005.

[165] E Pelosi, V Arena, A Skanjeti, V Pirro, A Douroukas, A Pupi et M Mancini. Role of whole-body 18F-choline PET/CT in disease detection in patients with biochemical relapse after radical treatment for prostate cancer. *La radiologia medica*, 113(6):895–904, 2008.

[166] Michael E Phelps. *PET : molecular imaging and its biological applications*. Springer, 2004.

[167] Michael E Phelps, Edward J Hoffman, SungCheng Huang et David E Kuhl. ECAT : a new computerized tomographic imaging system for positron-emitting radiopharmaceuticals. *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, 19(6): 635–647, 1978.

[168] Bernd J Pichler, Armin Kolb, Thomas Nägele et Heinz-Peter Schlemmer. PET/MRI : paving the way for the next generation of clinical multimodality imaging applications. *Journal of Nuclear Medicine*, 51(3):333–336, 2010.

[169] Bernd J Pichler, Hans F Wehrl et Martin S Judenhofer. Latest advances in molecular imaging instrumentation. *Journal of Nuclear Medicine*, 49(Suppl 2):5S–23S, 2008.

[170] G. Pratx, G. Chinn, P.D. Olcott et C.S. Levin. Fast, accurate and shift-varying line projections for iterative reconstruction using the GPU. *Medical Imaging, IEEE Transactions on*, 28(3):435–445, 2009. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4637849`.

[171] G. Pratx, P. Olcott, G. Chinn et C. Levin. Accelerated list-mode 3D-OSEM reconstruction for PET on a graphics processing unit. Dans *Society of Nuclear Medicine Annual Meeting Abstracts*, volume 47, page 183. Soc Nuclear Med, 2006. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4179464`.

[172] Guillem Pratx, Suleman Surti et Craig Levin. Fast list-mode reconstruction for time-of-flight PET using graphics hardware. *Nuclear Science, IEEE Transactions on*, 58(1): 105–109, 2011. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5635379`.

[173] Guillem Pratx et Lei Xing. GPU computing in medical physics : A review. *Medical physics*, 38:2685, 2011.

[174] Guillem Pratx et Lei Xing. GPU computing in medical physics : A review. *Medical physics*, 38(5):2685–2697, 2011.

[175] Jinyi Qi. Calculation of the sensitivity image in list-mode reconstruction for PET. *Nuclear Science, IEEE Transactions on*, 53(5):2746–2751, 2006.

[176] Jinyi Qi. Calculation of the sensitivity image in list-mode reconstruction for PET. *Nuclear Science, IEEE Transactions on*, 53(5):2746–2751, 2006.

[177] Jinyi Qi et R.H. Huesman. Propagation of errors from the sensitivity image in list mode reconstruction. *Medical Imaging, IEEE Transactions on*, 23(9):1094–1099, 2004.

[178] Jinyi Qi et Richard M Leahy. Iterative reconstruction techniques in emission computed tomography. *Physics in Medicine and Biology*, 51(15):R541, 2006.

[179] Jinyi Qi et Richard M Leahy. Iterative reconstruction techniques in emission computed tomography. *Physics in medicine and biology*, 51:R541, 2006. URL `http://iopscience.iop.org/0031-9155/51/15/R01`.

[180] Jinyi Qi, Richard M Leahy, Chinghan Hsu, Thomas H Farquhar et Simon R Cherry. Fully 3D Bayesian image reconstruction for the ECAT EXACT HR+. *Nuclear Science, IEEE Transactions on*, 45(3):1096–1103, 1998.

[181] Magdalena Rafecas, Brygida Mosler, Melanie Dietz, M Pogl, Alexandros Stamatakis, David P McElroy et Sibylle I Ziegler. Use of a Monte Carlo-based probability matrix for 3-D iterative reconstruction of MADPET-II data. *Nuclear Science, IEEE Transactions on*, 51(5):2597–2605, 2004.

[182] A. Raheja, T.F. Doniere et A.P. Dhawan. Multiresolution expectation maximization reconstruction algorithm for positron emission tomography using wavelet processing. *Nuclear Science, IEEE Transactions on*, 46(3):594–602, 1999. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=775585`.

[183] Arman Rahmim, Ju-Chieh Cheng, Stephan Blinder, Maurie-Laure Camborde et Vesna Sossi. Statistical dynamic image reconstruction in state-of-the-art high-resolution PET. *Physics in medicine and biology*, 50:4887, 2005. URL `http://iopscience.iop.org/0031-9155/50/20/010`.

[184] Arman Rahmim, M. Lenox, Andrew J Reader, C. Michel, Z. Burbar, Thomas J Ruth et Vesna Sossi. Statistical list-mode image reconstruction for the high resolution research tomograph. *Physics in Medicine and Biology*, 49(18):4239, 2004. URL `http://stacks.iop.org/0031-9155/49/i=18/a=004`.

[185] Arman Rahmim, J Tang, MA Lodge, Sahel Lashkari, Mohammad Reza Ay, R Lautamäki, BMW Tsui et FM Bengel. Analytic system matrix resolution modeling in PET : an application to Rb-82 cardiac imaging. *Physics in medicine and biology*, 53(21):5947, 2008.

[186] Arman Rahmim, Jing Tang et Habib Zaidi. Four-dimensional (4D) image reconstruction strategies in dynamic PET : beyond conventional independent frame reconstruction. *Medical physics*, 36:3654, 2009.

[187] MV Ranganath, A.P. Dhawan et N. Mullani. A multigrid expectation maximization reconstruction algorithm for positron emission tomography. *Medical Imaging, IEEE Transactions on*, 7(4):273–278, 1988. URL http://dx.doi.org/10.1109/42.14509.

[188] E Rapisarda, V Bettinardi, K Thielemans et MC Gilardi. Image-based point spread function implementation in a fully 3D OSEM reconstruction algorithm for PET. *Physics in medicine and biology*, 55(14):4131, 2010.

[189] A.J. Reader, R. Manavaki, S. Zhao, P.J. Julyan, D.L. Hastings et J. Zweit. Accelerated list-mode EM algorithm. *Nuclear Science, IEEE Transactions on*, 49(1):42–49, 2002. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=998679.

[190] Andrew J Reader, K. Erlandsson, Maggie A. Flower et R.J. Ott. Fast accurate iterative reconstruction for low-statistics positron volume imaging. *Physics in Medicine and Biology*, 43:835, 1998. URL http://iopscience.iop.org/0031-9155/43/4/012.

[191] Andrew J Reader, Julian C Matthews, Florent C Sureau, Claude Comtat, Régine Trebossen et Irène Buvat. Iterative kinetic parameter estimation within fully 4D PET image reconstruction. Dans *Nuclear Science Symposium Conference Record, 2006. IEEE*, volume 3, pages 1752–1756. IEEE, 2006.

[192] Andrew J Reader, Florent C Sureau, Claude Comtat, Régine Trébossen et Irene Buvat. Joint estimation of dynamic PET images and temporal basis functions using fully 4D ML-EM. *Physics in Medicine and Biology*, 51(21):5455, 2006.

[193] Andrew J Reader et Habib Zaidi. Advances in PET image reconstruction. *PET Clinics*, 2 (2):173–190, 2007.

[194] D. Ruijters, B.M. ter Haar Romeny et Paul Suetens. Efficient GPU-based texture interpolation using uniform B-splines. *Journal of Graphics, GPU, and Game Tools*, 13(4):61–69, 2008.

[195] JJ Scheins, Hans Herzog et N Jon Shah. Fully-3D PET image reconstruction using scanner-independent, adaptive projection data and highly rotation-symmetric voxel assemblies. *Medical Imaging, IEEE Transactions on*, 30(3):879–892, 2011.

[196] Jürgen J Scheins, Fritz Boschen et Hans Herzog. Analytical calculation of volumes-of-intersection for iterative, fully 3-D PET reconstruction. *Medical Imaging, IEEE Transactions on*, 25(10):1363–1369, 2006.

[197] M. Schellmann et S. Gorlatch. Comparison of two decomposition strategies for parallelizing the 3d list-mode OSEM algorithm. Dans *Proceedings Fully 3D Meeting and HPIR Workshop*, pages 37–40, 2007.

[198] M. Schellmann, S. Gorlatch, D. Meiländer, T. Kösters, K. Schäfers, F. Wübbeling et M. Burger. Parallel Medical Image Reconstruction : From Graphics Processors to Grids. *Parallel Computing Technologies*, pages 457–473, 2009. URL `http://www.springerlink.com/content/e722577307p15pm5/`.

[199] M. Schellmann, J. Vörding et S. Gorlatch. Systematic parallelization of medical image reconstruction for graphics hardware. *Euro-Par 2008–Parallel Processing*, pages 811–821, 2008. URL `http://www.springerlink.com/content/m583180647215123/`.

[200] Holger Scherl, Benjamin Keck, Markus Kowarschik et Joachim Hornegger. Fast GPU-based CT reconstruction using the common unified device architecture (CUDA). Dans *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, volume 6, pages 4464–4466. IEEE, 2007.

[201] C Ross Schmidtlein, Assen S Kirov, Sadek A Nehmeh, Yusuf E Erdi, John L Humm, Howard I Amols, Luc M Bidaut, Alex Ganin, Charles W Stearns, David L McDaniel et al. Validation of GATE Monte Carlo simulations of the GE Advance/Discovery LS PET scanners. *Medical physics*, 33:198, 2006.

[202] C. Schretter. Event-by-event image reconstruction from list-mode PET data. *Image Processing, IEEE Transactions on*, 18(1):117–124, 2008.

[203] Vitali V Selivanov et Roger Lecomte. Fast PET image reconstruction based on SVD decomposition of the system matrix. *Nuclear Science, IEEE Transactions on*, 48(3):761–767, 2001.

[204] Vitali V Selivanov, Martin D Lepage et Roger Lecomte. List-mode image reconstruction for real-time PET imaging. *Journal of Visual Communication and Image Representation*, 17(3):630–646, 2006.

[205] L.A. Shepp et Y. Vardi. Maximum likelihood reconstruction for emission tomography. *Medical Imaging, IEEE Transactions on*, 1(2):113–122, 1982.

[206] Miho Shidahara, Charalampos Tsoumpas, CJ McGinnity, Takashi Kato, Hajime Tamura, Alexander Hammers, Hiroshi Watabe et FE Turkheimer. Wavelet-based resolution recovery using an anatomical prior provides quantitative recovery for human population phantom PET [11C] raclopride data. *Physics in medicine and biology*, 57(10):3107, 2012.

[207] S Shokouhi, P Vaska, S Southekal, D Schlyer, M Purschke, V Dzordzhadze, C Woody, S Stoll, DL Alexoff, D Rubins et al. Statistical 3D image reconstruction for the RatCAP PET tomograph using a physically accurate, Monte Carlo based system matrix. Dans *Nuclear Science Symposium Conference Record, 2004 IEEE*, volume 6, pages 3901–3905. IEEE, 2004.

[208] AK Shukla et Utham Kumar. Positron emission tomography : An overview. *Journal of medical physics/Association of Medical Physicists of India*, 31(1):13, 2006.

[209] Robert L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics*, 12(2):252–255, 1985. URL `http://link.aip.org/link/?MPH/12/252/1`.

[210] Donald L Snyder et Michael I Miller. The use of sieves to stabilize images produced with the EM algorithm for emission tomography. *Nuclear Science, IEEE Transactions on*, 32 (5):3864–3872, 1985.

[211] Sangeetha Somayajula, Christos Panagiotou, Anand Rangarajan, Quanzheng Li, Simon R Arridge et Richard M Leahy. PET image reconstruction using information theoretic anatomical priors. *Medical Imaging, IEEE Transactions on*, 30(3):537–549, 2011.

[212] Steven Staelens, Yves D'Asseler, Stefaan Vandenberghe, Michel Koole, Ignace Lemahieu et Rik Van de Walle. A three-dimensional theoretical model incorporating spatial detection uncertainty in continuous detector PET. *Physics in medicine and biology*, 49(11):2337, 2004.

[213] Suleman Surti, Austin Kuhn, Matthew E Werner, Amy E Perkins, Jeffrey Kolthammer et Joel S Karp. Performance of Philips Gemini TF PET/CT scanner with special consideration for its time-of-flight imaging capabilities. *Journal of Nuclear Medicine*, 48(3): 471–480, 2007.

[214] Akos Szlávecz, Gábor Hesz, Tamás Bükki, Béla Kári et Balázs Benyó. GPU-based acceleration of the MLEM algorithm for SPECT parallel imaging with attenuation correction and compensation for detector response. Dans *Proceedings of the 18th IFAC World Congress. Milan, Italy., August*, volume 28, pages 6195–6200, 2011.

[215] Jing Tang et Arman Rahmim. Bayesian PET image reconstruction incorporating anato-functional joint entropy. *Physics in medicine and biology*, 54(23):7063, 2009.

[216] Michel M Ter-Pogossian, Michael E Phelps, Edward J Hoffman et Nizar A Mullani. A positron-emission transaxial tomograph for nuclear imaging (PETT). *Radiology*, 114(1): 89–98, 1975.

[217] J Thibaud. L'annihilation des positrons au contact de la matiere et la radiation qui en résulte. *CR Acad. Sci*, 197:1629–1632, 1933.

[218] K. Thielemans, S. Mustafovic et C. Tsoumpas. STIR : Software for tomographic image reconstruction release 2. Dans *Nuclear Science Symposium Conference Record*, volume 4, pages 2174–2176. IEEE, 2007. ISBN 1424405602.

[219] Kris Thielemans, Charalampos Tsoumpas, Sanida Mustafovic, Tobias Beisel, Pablo Aguiar, Nikolaos Dikaios et Matthew W Jacobson. STIR : software for tomographic image reconstruction release 2. *Physics in medicine and biology*, 57(4):867, 2012.

[220] Zhen Tian, Xun Jia, Kehong Yuan, Tinsu Pan et Steve B Jiang. Low-dose CT reconstruction via edge-preserving total variation regularization. *Physics in medicine and biology*, 56(18):5949, 2011.

[221] Michel S Tohme et Jinyi Qi. Iterative image reconstruction for positron emission tomography based on a detector response function estimated from point source measurements. *Physics in medicine and biology*, 54(12):3709, 2009.

[222] David W Townsend, Thomas Beyer et Todd M Blodgett. PET/CT scanners : a hardware approach to image fusion. Dans *Seminars in nuclear medicine*, volume 33, pages 193–204. Elsevier, 2003.

[223] DW Townsend, A Geissbuhler, M Defrise, EJ Hoffman, TJ Spinks, DL Bailey, M-C Gilardi et T Jones. Fully three-dimensional reconstruction for a PET camera with retractable septa. *Medical Imaging, IEEE Transactions on*, 10(4):505–512, 1991.

[224] Charalampos Tsoumpas, Federico E Turkheimer et Kris Thielemans. A survey of approaches for direct parametric image reconstruction in emission tomography. *Medical physics*, 35:3963, 2008.

[225] Roel Van Holen, Brian W Miller, Jared W Moore, Stefaan Vandenberghe et Harrison H Barrett. Object-space interpolation of SPECT system matrices from point-source measurements. Dans *11th International meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine (Fully 3D 11)*, pages 419–422, 2011.

[226] Stefaan Vandenberghe, Steven Staelens, Charles L Byrne, Edward J Soares, Ignace Lemahieu et Stephen J Glick. Reconstruction of 2D PET data with Monte Carlo generated system matrix for generalized natural pixels. *Physics in medicine and biology*, 51 (12):3105, 2006.

[227] Andrea Varrone, Nils Sjholm, Lars Eriksson, Balazs Gulys, Christer Halldin et Lars Farde. Advancement in PET quantification using 3D-OP-OSEM point spread function reconstruction with the HRRT. *European Journal of Nuclear Medicine and Molecular Imaging*, 36:1639–1650, 2009. URL `http://dx.doi.org/10.1007/s00259-009-1156-3`. ISSN 1619-7070.

[228] P Vaska, C Woody, D Schlyer, J-F Pratte, S Junnarkar, S Southekal, S Stoll, D Schulz, W Schiffer, D Alexoff et al. The design and performance of the 2 nd-generation RatCAP awake rat brain PET system. Dans *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, volume 6, pages 4181–4184. IEEE, 2007.

[229] Jeroen Verhaeghe, Yves D'Asseler, Stefaan Vandenberghe, Steven Staelens, Rik Van de Walle et Ignace Lemahieu. ML reconstruction from dynamic list-mode PET data using temporal splines. Dans *Nuclear Science Symposium Conference Record, 2004 IEEE*, volume 5, pages 3146–3150. IEEE, 2004.

[230] Jeroen Verhaeghe, Dimitri Van De Ville, Ildar Khalidov, Yves D'Asseler, Ignace Lemahieu et Michael Unser. Dynamic PET reconstruction using wavelet regularization with adapted basis functions. *Medical Imaging, IEEE Transactions on*, 27(7):943–959, 2008.

[231] M D Walker, M-C Asselin, P.J. Julyan, M. Feldmann, P S Talbot, T. Jones et J C Matthews. Bias in iterative reconstruction of low-statistics PET data : benefits of a resolution model. *Physics in Medicine and Biology*, 56(4):931, 2011. URL `http://stacks.iop.org/0031-9155/56/i=4/a=004`.

[232] Matthew D Walker, MC Asselin, Peter J Julyan, M Feldmann, PS Talbot, T Jones et JC Matthews. Bias in iterative reconstruction of low-statistics PET data : benefits of a resolution model. *Physics in Medicine and Biology*, 56(4):931, 2011.

[233] Guobao Wang et Jinyi Qi. Generalized algorithms for direct reconstruction of parametric images from dynamic PET data. *Medical Imaging, IEEE Transactions on*, 28(11):1717–1726, 2009.

[234] W Wang, Z Hu, EE Gualtieri, MJ Parma, ES Walsh, D Sebok, Y-L Hsieh, C-H Tung, X Song, JJ Griesmer et al. Systematic and distributed time-of-flight list mode PET reconstruction. Dans *Nuclear Science Symposium Conference Record, 2006. IEEE*, volume 3, pages 1715–1722. IEEE, 2006.

[235] Zigang Wang, Guoping Han, Tianfang Li et Zhengrong Liang. Speedup OS-EM image reconstruction by PC graphics card technologies for quantitative SPECT with varying focal-length fan-beam collimation. *Nuclear Science, IEEE Transactions on*, 52(5):1274–1280, 2005.

[236] Otto Warburg, Franz Wind et Erwin Negelein. The metabolism of tumors in the body. *The Journal of general physiology*, 8(6):519–530, 1927.

[237] Miles N Wernick et John N Aarsvold. *Emission tomography : the fundamentals of PET and SPECT*. Access Online via Elsevier, 2004.

[238] Miles N Wernick et John N Aarsvold. *Emission tomography : the fundamentals of PET and SPECT*. Access Online via Elsevier, 2004.

[239] Miles N Wernick, E James Infusino et Milos Milosevic. Fast spatio-temporal image reconstruction for dynamic PET. *Medical Imaging, IEEE Transactions on*, 18(3):185–195, 1999.

[240] Marinke Westerterp, Jan Pruim, Wim Oyen, Otto Hoekstra, Anne Paans, Eric Visser, Jan van Lanschot, Gerrit Sloof et Ronald Boellaard. Quantification of FDG PET studies using standardised uptake values in multi-centre trials : effects of image reconstruction, resolution and ROI definition parameters. *European journal of nuclear medicine and molecular imaging*, 34(3):392–404, 2007.

[241] K Wienhard, M Schmand, ME Casey, K Baker, J Bao, L Eriksson, WF Jones, C Knoess, M Lenox, M Lercher et al. The ECAT HRRT : performance and first clinical application of the new high resolution research tomograph. *Nuclear Science, IEEE Transactions on*, 49(1):104–110, 2002.

[242] Donald W Wilson, Benjamin MW Tsui et Harrison H Barrett. Noise properties of the EM algorithm. II. Monte Carlo simulations. *Physics in medicine and biology*, 39(5):847, 1994.

[243] Frank R Wrenn, Myron L Good et Philip Handler. The use of positron-emitting radioisotopes for the localization of brain tumors. *Science*, 113(2940):525–527, 1951.

[244] Fang Xu et Klaus Mueller. Real-time 3D computed tomographic reconstruction using commodity graphics hardware. *Physics in medicine and biology*, 52(12):3405, 2007.

[245] Peiliang Xu. Truncated SVD methods for discrete linear ill-posed problems. *Geophysical Journal International*, 135(2):505–514, 1998.

[246] Wei Xu et Klaus Mueller. Learning effective parameter settings for iterative CT reconstruction algorithms. Dans *Workshop on High Performance Image Reconstruction (HPIR)*, 2009.

[247] Xiao-Liang Xu, Jeih-San Liow et Stephen C Strother. Iterative algebraic reconstruction algorithms for emission computed tomography : A unified framework and its application to positron emission tomography. *Medical physics*, 20:1675, 1993.

[248] Yuan Xu, Ti Bai, Hao Yan, Luo Ouyang, Jing Wang, Arnold Pompos, Linghong Zhou, Steve Jiang et Xun Jia. Ultrafast cone-beam CT scatter correction with GPU-based Monte Carlo simulation. *International Journal of Cancer Therapy and Oncology*, 2(2), 2014.

[249] Jianhua Yan, Beata Planeta-Wilson et Richard E Carson. Direct 4D list mode parametric reconstruction for PET with a novel EM algorithm. Dans *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, pages 3625–3628. IEEE, 2008.

[250] Zhou Yu, J-B Thibault, Charles A Bouman, Ken D Sauer et Jiang Hsieh. Fast model-based X-ray CT reconstruction using spatially nonhomogeneous ICD optimization. *Image Processing, IEEE Transactions on*, 20(1):161–175, 2011.

[251] Bahi Z, Bert J, Autret A et Visvikis D. High Performance Multi-GPU Acceleration for Fully 3D List-Mode PET Reconstruction. Dans *IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, pages 3390–3393. IEEE, 2012.

[252] Habib Zaidi et Marie-Louise Montandon. Scatter compensation techniques in PET. *PET Clinics*, 2(2):219–234, 2007.

[253] Long Zhang, Steven Staelens, Roel Van Holen, Jan De Beenhouwer, Jeroen Verhaeghe, Iwan Kawrakow et Stefaan Vandenberghe. Fast and memory-efficient Monte Carlo-based image reconstruction for whole-body PET. *Medical Physics*, 37(7):3667–3676, 2010. URL `http://link.aip.org/link/?MPH/37/3667/1`.

[254] Huaxia Zhao et Andrew J Reader. Fast ray-tracing technique to calculate line integral paths in voxel arrays. Dans *Nuclear Science Symposium Conference Record, 2003 IEEE*, volume 4, pages 2808–2812. IEEE, 2003.

[255] Xing Zhao, Jing-jing Hu et Peng Zhang. GPU-based 3D cone-beam CT image reconstruction for large data volume. *Journal of Biomedical Imaging*, 2009:8, 2009.

[256] Jian Zhou et Jinyi Qi. Fast and efficient fully 3D PET image reconstruction using sparse system matrix factorization with GPU acceleration. *Physics in Medicine and Biology*, 56(20):6739, 2011. URL `http://stacks.iop.org/0031-9155/56/i=20/a=015`.

[257] Zhenyu Zhou, RN Leahy et Jinyi Qi. Approximate maximum likelihood hyperparameter estimation for Gibbs priors. *Image Processing, IEEE Transactions on*, 6(6):844–861, 1997.