



REPRÉSENTATION D'IMAGES 3D POUR LES SIMULATIONS D'INTERACTIONS RAYONNEMENT MATIÈRE EN RADIOTHÉRAPIE

Rapport de stage
Février - Juin 2005

Présenté par
Antoine VACAVANT
Master Recherche Informatique

Encadrants
David SARRUT, Serge MIGUET

Laboratoire d'InfoRmatique en Images et Systèmes d'information - UMR 5205
Université Claude Bernard Lyon 1
Bâtiment Nautibus
43, boulevard du 11 Novembre 1918
69 622 Villeurbanne cedex

Centre Régional de Lutte Contre le Cancer Léon-Bérard
28, rue Laënnec
69008 LYON

Remerciements

Je voudrais d'abord remercier David Sarrut pour m'avoir accepté en stage au CLB. Lui et son équipe de recherche sont très compétents et m'ont apporté beaucoup de connaissances en Informatique et en Physique médicale. Je note au passage que les médecins ont été très pédagogues et m'ont enseigné des notions intéressantes, alors que je n'ai que peu de formation dans ce domaine. L'accueil a été très chaleureux et l'ambiance générale est très bonne. Ce stage a été instructif et agréable. Je remercie toutes les personnes qui m'ont aidé pour le stage et celles qui ont lu mon rapport et m'ont donné de bons conseils.

En bref, je remercie

★ mes encadrants :

David Sarrut et Serge Miguet

★ les doctorants de l'équipe :

Vlad Boldea, Simon Rit, Jean-Noël Badel et Myriam Ayadi

★ les stagiaires :

Nathalie Dufour, Nicolas Curtet et Elsa Giroud

★ pour son aide en géométrie discrète et algorithmique :

David Coeurjolly

★ les personnes qui m'ont aidé dans la rédaction de ce rapport :

David Sarrut, Vlad Boldea, Simon Rit, Serge Miguet et David Coeurjolly



Résumé

En radiothérapie, la planification des traitements nécessite une simulation préalable des séances d'irradiation pour observer et analyser les informations balistiques du rayonnement (photons de haute énergie). On vérifie alors que la tumeur a été suffisamment irradiée, et que les tissus sains ont été épargnés pendant la séance. **Geant4** est une bibliothèque C++ de simulation de Monte Carlo des hautes énergies que nous avons adapté, puis amélioré afin qu'il prenne en compte une cible modélisant un patient sous la forme d'une matrice 3D de voxels, initialement obtenue par un examen scanner tomodensitométrique. Nous avons modifié **Geant4** pour qu'il prenne en compte la nature régulière d'une matrice 3D. Cette structure régulière présente l'inconvénient de répéter les données inutilement. Si l'on classe les voxels de même intensité pour créer ensuite une grille irrégulière isothétique, le nombre de volumes pour décrire le patient et les temps de simulation diminuent. Nous avons appliqué ce principe en codant dans un premier temps l'image avec un RLE (*Run Length Encoding*) pour compresser les données de l'image. La géométrie du patient est plus compacte et le système de voisinage reste simple. Les temps de simulation sont sensiblement réduits, et nous garantissons une précision satisfaisante pour nos simulations ainsi configurées grâce à un comparatif entre les cartes de dose calculées et une carte issue de simulations références basées sur une grille régulière classique. Dans l'avenir, d'autres approches que le RLE seront développées.

Mots-clés

Simulations Monte Carlo, imagerie médicale, traitements d'images, géométrie discrète.

Abstract

In radiotherapy, treatments planning need simulations of radiation sessions to observe and analyze targetting information of irradiation beam (high energy photons). Physicists can check if the tumor was irradiated enough during the session, and if healthy tissues were avoided. **Geant4** is a C++ library for Monte Carlo simulation of high energy. We have adapted and improved it so that it takes into account a target modelizing the patient in a three-dimension matrix, obtained from a computerized tomography scan. We modified **Geant4** to consider the regularity of a 3D matrix. However, this regular structure unnecessarily repeats image data. If we group voxels of same intensity to create a irregular isothetic grid, the volumes number to represent the patient and the simulation time decrease. At first, we applied this principle by coding the image with a RLE (Run Length Encoding) to compress the image data. The patient geometry is more compact and the neighborhood system is simple. The time of simulations is greatly reduced, and we guarantee a satisfying precision for so configured simulations, comparing computed dose maps and a map deduced from a gold standard simulation conducted on a regular grid. In the future, other approaches than RLE will be developed.

Keywords

Monte Carlo simulations, medical imaging, image processing, discrete geometry.

Table des matières

Introduction	3
1 Contexte	5
1.1 Simulations de Monte Carlo	5
1.2 Simulation physique de radiothérapie	6
1.3 Objectifs	8
2 État de l’art	9
2.1 Conversion d’un scan CT au format DICOM en un objet 3D pour la simulation	9
2.2 Simulation physique de radiothérapie avec Geant4	10
3 Représentation d’images 3D dans une simulation physique de Monte Carlo	11
3.1 Segmentation	11
3.2 Simulations sur une grille 3D régulière	11
3.2.1 Processus de simulation dans Geant4	11
3.2.2 Modélisation géométrique	13
3.2.3 Navigation dans la géométrie	14
3.2.4 Notre approche	15
3.3 Vers une simulation sur une grille 3D irrégulière isothétique	15
3.3.1 Notion de grille irrégulière isothétique	15
3.3.2 Description d’une grille irrégulière isothétique simple	17
3.3.3 Bilan algorithmique de notre approche	18
4 Expérimentations et résultats obtenus	20
4.1 Matériel et méthode	20
4.1.1 Images de test	20
4.1.2 Simulation de référence	20
4.1.3 Méthodes testées	22
4.2 Résultats	23
4.2.1 Observations générales	24
4.2.2 Comparaison détaillée	25
Bilan et perspectives	28
Références	29
Annexe A : Résultats des simulations tests	31
Annexe B : Différences observées par matériau	33

Introduction

Dans le cadre du Master Recherche Informatique à l'Université Claude Bernard Lyon 1, j'ai effectué mon stage au Centre Léon Bérard (CLB, Lyon, France). C'est une structure privée à but non lucratif participant au service de santé publique dans la lutte contre le cancer. C'est aussi un centre de recherche et d'enseignement. Il compte aujourd'hui douze départements dont celui de Radiothérapie, où l'on prodigue deux mille traitements par an. Le groupe de recherche dans lequel j'ai évolué est une équipe multidisciplinaire Informatique-Physique-Médecine composée de chercheurs appartenant à trois UMR CNRS : LIRIS, CREATIS et IPNL. Pour ma part, j'appartenais à la partie CLB-LIRIS, localisée au CLB. Mon stage s'inscrit dans un projet de modélisation et d'acquisition d'images 4D, pour mieux prendre en compte le mouvement respiratoire lors de l'acquisition d'images et pendant les traitements par radiothérapie.

La radiothérapie est une technique de traitement du cancer dont l'objectif consiste à délivrer des rayons ionisants de haute énergie sur une zone cancéreuse afin de la stériliser, tout en préservant au maximum les tissus sains. Dans ce type de traitement, la planification requiert une simulation précise des séances d'irradiation. Dans un premier temps, un physicien médical contourne la tumeur, puis intègre la région sélectionnée avec le positionnement du patient et les paramètres du faisceau (énergie, position, largeur, etc.) dans le simulateur. Les simulations permettent ensuite d'observer quels sont les tissus (sains ou non) atteints par les rayons incidents. Pour corriger la balistique calculée, on modifie les paramètres d'irradiation (positionnement et faisceau) pour viser au mieux la tumeur.

Les logiciels de simulations du dépôt de dose généralement utilisés s'exécutent rapidement (quelques secondes) mais simplifient les interactions entre les particules et les matériaux représentés au sein du patient. Ces approximations sont préjudiciables notamment en présence d'hétérogénéités (proximité de matières fortement et faiblement denses, par exemple l'os et les poumons). Au Centre Léon Bérard, les tumeurs cancéreuses pulmonaires sont souvent irradiées alors que le patient est en apnée contrôlée (à l'aide d'un système nommé ABC ou *Active Breathing Control*). Cette technique est pénible pour les patients, subissant généralement d'autres traitements complémentaires fatigants (chimiothérapie, etc.). Pour traiter en respiration libre, il est nécessaire de simuler la ventilation pulmonaire du patient pendant la simulation, afin de prédire les effets de l'irradiation sur les poumons en mouvement.

Les codes de simulation dits de Monte Carlo permettent de simuler la trajectoire des particules sortant de l'accélérateur en décrivant pas à pas leurs caractéristiques (énergie, vitesse, masse, etc.), et l'énergie qu'elles déposent éventuellement dans les tissus du patient. La modélisation physique est très poussée et permet de simuler avec précision les interactions des particules avec leur environnement. L'inconvénient majeur de ces logiciels est de prendre un temps important pour exécuter des simulations complexes. **Geant4**, réalisé par un consortium d'équipes internationales (notamment le CERN à Genève, Suisse), est une bibliothèque C++ de simulations de hautes énergies qui est facilement adaptable à des applications complexes en radioprotection, en calorimétrie, et en physique médicale. Ce code de simulation est actuellement un des plus puissants dispo-

nibles pour la communauté scientifique, car il modélise la plupart des connaissances en physique des particules.

A long terme, l'objectif principal du projet dans lequel s'inscrit cette étude est de calculer le dépôt de dose dans des cibles complexes et dynamiques. Cela implique une modélisation 3D+t du patient ainsi que la création de simulations compatibles avec un tel modèle.

Une tâche importante de ce projet est la considération des aspects physiques et géométriques des simulations physiques. Je me suis penché plus précisément sur la géométrie représentée dans **Geant4**, en cernant simplement les propriétés physiques des particules simulées avec l'aide de physiciens du CLB.

L'objectif de ce stage est de modéliser un patient, décrit initialement par un scan CT au format DICOM, en une image 3D transmise ensuite à **Geant4**. L'image est décrite par des voxels (de dimension $2 \times 2 \times 2 \text{ mm}^3$ par exemple), au sein desquels on veut connaître le dépôt d'énergie. Cependant, le nombre de voxels est très important (dans la même résolution, environ neuf millions de voxels) et ralentit le calcul des trajectoires des particules au sein du patient. De plus, l'occupation mémoire est lourde. Notre travail a donc consisté à proposer des méthodes d'une part pour adapter le logiciel pour qu'il prenne en compte des images de plusieurs millions de voxels, d'autre part pour accélérer le temps de calcul.

Nous détaillerons d'abord le contexte dans lequel s'inscrit cette étude, et notamment les principes physiques et informatiques autour desquels notre travail s'articule.

Nous montrerons ensuite, dans la partie 2, les travaux traitant des simulations physiques sur une structure voxelisée.

La partie 3 illustre ma contribution dans le projet décrit plus haut. Elle regroupe les avancées dans la simulation physique sur une grille régulière de voxels avec **Geant4**, puis sur une grille irrégulière isothétique.

Enfin, nous montrons les résultats obtenus sur des critères de rapidité et de qualité des modèles géométriques que nous avons réalisés, sur différentes résolutions d'image 3D.

1 Contexte

1.1 Simulations de Monte Carlo

Pour modéliser et simuler des phénomènes physiques complexes, on remarque souvent l'emploi de méthodes de Monte Carlo. Cette technique sert initialement à chercher les solutions de problèmes mathématiques spécifiques (calcul d'intégrales, etc.) par échantillonnage stochastique [3]. Plus précisément, on cherche à approximer le comportement d'un phénomène, c'est à dire la probabilité qu'il intervienne, grâce à la génération de nombres aléatoires. On peut également simuler un modèle physique particulière (photons, etc.) grâce à cette approche.

Les méthodes de Monte Carlo s'appliquent aujourd'hui dans des domaines divers. Par exemple, pour créer des scènes de synthèse, on simule le comportement des photons sur la surface des objets présents dans la scène lors d'un lancer de rayons. Les rayons lumineux indirects sont tracés grâce à cette technique [21]. En vision artificielle, le suivi temporel d'objets peut être assuré par un filtre à particules, qui se base sur des échantillonnages de Monte Carlo [18]. Dans le cadre de la chirurgie de l'os assistée par ordinateur, on peut citer une technique originale de recalage rigide 3D/3D, basé surface dans [13]. Elle permet de guider le chirurgien pendant l'opération, en recalant la planification d'intervention qu'il a déterminé au préalable sur un scan CT.

Dans le domaine de la physique des particules qui nous concerne, le code de Monte Carlo permet de modéliser la trajectoire de particules élémentaires. Par exemple on simule des particules γ (photon gamma) en radiothérapie conventionnelle, et des ions carbonés ou des protons en hadronthérapie.

Le principe de base est de simuler, pour tout temps t , la réaction de la particule en considérant ses caractéristiques propres (direction, vitesse, type de particule, masse, etc.) et l'environnement dans lequel elle évolue (matériau, pression, température, champ électromagnétique, etc.). On détermine alors si la particule va simplement se déplacer, créer d'autres particules, ou encore "mourir" (c'est à dire sortir définitivement de la zone d'étude). Le nombre et les types de phénomènes physiques simulés sont fixés à l'avance. Voici quelques exemples de ces phénomènes [22] :

- L'effet photo-électrique se manifeste par l'éjection d'un électron e^- de la matière après absorption d'un photon par cette dernière.
- La diffusion Compton intervient par émission d'un photon provenant d'un électron attaché à son noyau atomique.
- La production de paire e^+/e^- par un photon de haute énergie.

La simulation est étudiée pas à pas. Ainsi le parcours de la particule en un laps de temps dt est appelé *step*. À l'issue de chaque *step*, on cherche à connaître le comportement de la particule et ses caractéristiques (figure 1).

L'algorithme de simulation peut se décomposer ainsi [3] :

1. Au départ d'un *step*, on détermine quel est le processus physique qui va intervenir (photo-électrique, Compton, production de paire, etc.). Pour ce faire :

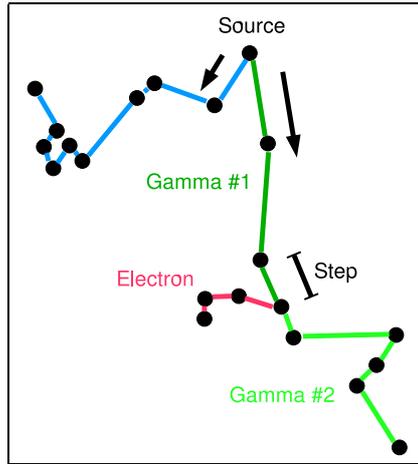


FIG. 1: Exemple d'annihilation d'une particule source γ dans de l'eau (#1). Elle crée un électron e^- et une particule γ secondaire (#2). Le step représente une unité de temps dt .

- ▶ On se réfère d'abord aux caractéristiques de la particule. On calcule alors pour chaque processus le NMFP (*Number of Mean Free Path*) grâce à une loi exponentielle d'interaction. C'est ici que l'on génère des nombres aléatoires pour connaître les NMFP.
 - ▶ Suivant le matériau dans lequel se situe la particule (eau, air, etc.), on convertit chacun de ces NMFP en longueurs physiques (PL). La PL représente la distance que parcourt la particule après le processus associé. On sélectionne alors celui de plus petite PL , notée $\min(PL)$.
2. Si la particule appartient à un objet de la scène et que la distance au bord de cet objet (GL) est inférieure à $\min(PL)$, alors on ne considère pas le phénomène choisi précédemment. On transporte la particule au bord de l'objet, puis on passe à l'étape 1 ¹.
 3. Sinon, on applique le processus et on transporte la particule sur la distance $\min(PL)$ déterminée à l'étape 1.
 4. Si la particule est toujours en vie, on boucle à l'étape 1.
 5. Sinon, on arrête.

Il faut remarquer que l'étape 1 de cet algorithme est le coeur de la simulation physique, où l'on détermine le processus qui doit intervenir grâce à des nombres aléatoires. L'étape 2 est très dépendante de la structure géométrique de la scène ; la modélisation du patient choisie influence donc considérablement cette phase de l'algorithme.

1.2 Simulation physique de radiothérapie

Notre but est de modéliser le trajet de chaque particule γ dans la scène composée du patient ou du fantôme à étudier. Lorsqu'une particule interagit (quel que soit le processus observé) au sein du patient ou d'un objet détecteur (imageur portal par exemple), on

¹On remarquera que la particule appartient désormais au volume suivant (à la frontière) sous **Geant4**.

considère l'énergie qu'elle libère à cet endroit (nommée aussi *edep* pour énergie déposée). Cette énergie représente aussi les dégâts occasionnés par cette particule dans les tissus qu'elle a rencontré. La dosimétrie *in vivo* est l'étude du dépôt de dose (*edep* cumulée) dans le patient. La figure 2 illustre le schéma global de notre étude.

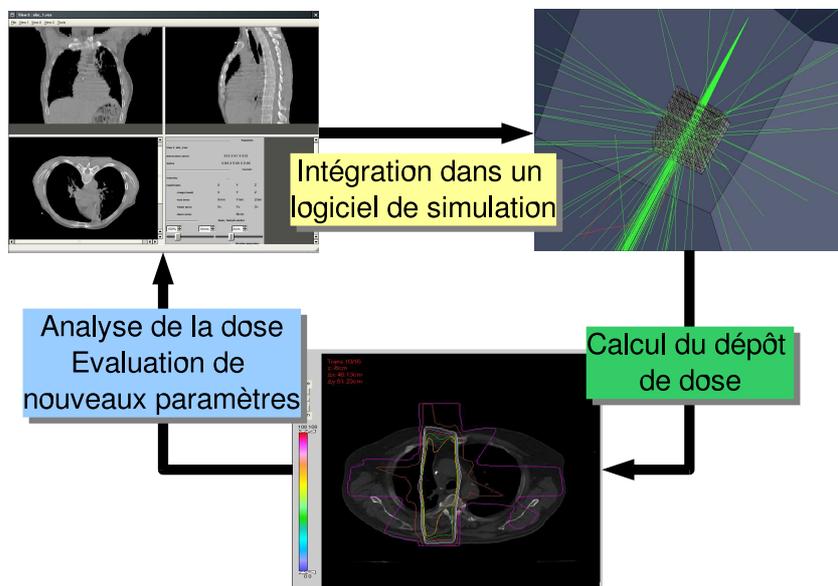


FIG. 2: Le système global de planification et de simulation de radiothérapie. A partir d'un scan CT, on initialise les paramètres de la séance (position du patient, rayon, etc.), puis on intègre ces éléments dans *Geant4*. A partir de cette simulation, on examine la qualité de la séance grâce à la carte de dose. Cette analyse permet donc de modifier la configuration du traitement, et donc de la simuler encore jusqu'à ce qu'une validation (d'un médecin ou d'un physicien médical) autorise la séance réelle sous les conditions appropriées.

Nous désirons représenter le patient sous la forme d'une matrice 3D, à partir d'un scan CT. Dans ce cas, le dépôt de dose peut être relevé au sein de chaque voxel de la matrice. On peut savoir quel a été l'*edep* dans chaque volume de $1 \times 1 \times 1 \text{ mm}^3$ par exemple. Une telle image contient initialement plusieurs millions de voxels, ce qui signifie que les particules changeront fréquemment de volume au sein de la matrice. Comme la structure géométrique influence la simulation et ses temps de calcul, il faut choisir une structure de données efficace pour représenter la matrice 3D dans *Geant4*.

L'auteur de [11] s'est proposé d'implanter différentes structures de données (octree, BSP ou *Binary Space Partitioning*, etc.) pour souligner celles qui accélèrent visiblement la méthode du lancer de rayons. Il faut remarquer que cet algorithme est comparable à notre processus de simulation des particules.

- Le phénomène physique simulé est discrétisé. Les sources émettent des particules de différents types dans notre étude, alors qu'il s'agit de rayons lumineux (et éventuellement des photons dans certaines méthodes avancées) dans le lancer de rayons ;
- on calcule la trajectoire de particules dans une scène en trois dimensions en simulant

leur comportement vis à vis du matériau dans lequel elle évolue, comme les rayons lumineux qui changent de direction suivant la matière des objets qu'ils rencontrent ;

- dans les deux domaines, la réduction des temps de calcul est un problème toujours d'actualité.

Par contre, notre *steptracing* se comporte comme un *raytracing* séquentiel stochastique.

- Aucune information globale sur le parcours de la particule n'est calculable (particule qui passe à côté du patient, etc.) et on simule pas à pas son comportement ;
- chaque nouvelle trajectoire ne peut se prédire à l'avance, lors d'un nouveau *step*, la direction que va prendre la particule est calculée par un processus aléatoire.

Cette étude nous permet d'aborder le problème de la représentation géométrique du patient dans **Geant4**.

1.3 Objectifs

Le premier objectif de ce stage est de représenter le patient grâce à une matrice 3D régulière. Il est clair que la précision et le temps de la simulation dépendent

- de la manière dont on simule les phénomènes physiques, plus précisément la génération des nombres aléatoires et le calcul de la longueur physique de chaque processus (étape 1 de l'algorithme de simulation) ;
- de la résolution de cette grille, puisque les particules seront bloquées sur un grand nombre de *steps* (étape 2). Par exemple, pour une image de résolution $5 \times 5 \times 5 \text{ mm}^3$, constituée de 250 000 voxels, et 1 million de particules générées, on observe près de 80 millions de *steps* au total. Nous avons calculé que les *steps* aboutissant effectivement à un processus physique en représentent 10 %, alors que les particules sont bloquées par la géométrie sur 90% des *steps* au sein de la matrice.

Nous avons collaboré avec des physiciens pour fixer et paramétrer les phénomènes qui peuvent intervenir dans nos simulations de radiothérapie. Cette étude devrait être approfondie pour produire une simulation plus précise.

Dans ce stage, nous nous sommes concentrés sur le deuxième facteur d'accélération pour optimiser le compromis entre précision de calcul du dépôt de dose et temps de simulation. Dans **Geant4**, les classes de construction géométriques ne prévoient pas initialement l'implantation d'une grille 3D régulière. A notre connaissance, seules 5 publications récentes traitent de simulations de radiothérapie supportées par une grille régulière de plusieurs millions de voxels. Nous proposons dans la partie 3.2 une approche pour représenter efficacement une telle grille dans **Geant4**.

De plus, comme les matrices que nous cherchons à manipuler sont volumineuses, nous détaillons l'emploi des grilles irrégulières isothétiques (dont les cellules 3D sont des parallélépipèdes et chacune d'entre elles peut avoir des dimensions différentes) dans la partie 3.3, afin d'accélérer les simulations. Nous n'avons trouvé dans la littérature aucune mention aux grilles irrégulières dans les simulations de radiothérapie.

2 État de l’art

La simulation de radiothérapie par des codes de Monte Carlo, et plus particulièrement *Geant4*, est un domaine de recherche récent sur lequel on dénombre encore peu de parutions d’articles. Certaines études se concentrent sur la segmentation de l’image en classes de matériaux (os, graisse, etc.) afin de la représenter précisément ensuite dans un simulateur [17, 16, 19]. D’autres décrivent la modélisation 3D ou 4D sur laquelle repose la simulation [15, 12].

2.1 Conversion d’un scan CT au format DICOM en un objet 3D pour la simulation

La conversion du scan du patient en une image 3D demande plusieurs traitements qui déterminent la précision de la simulation, mais aussi sa vitesse d’exécution. La phase la plus importante est la segmentation de l’image en différentes classes de matériaux (eau, air, graisse, etc.). En effet, on a en entrée une image au format DICOM dont les intensités CT s’étalent entre -1000 et 2000 *HU* (Unités Hounsfield) ², et on veut en déduire une image avec un nombre de classes très petit (comparativement aux 3000 valeurs possibles initialement) ; ce nombre de groupes de matériaux est généralement de l’ordre de la dizaine [16, 19]. On peut lire dans [17] que la définition de l’image initiale ne permet pas d’aller au delà de quelques dizaines de classes.

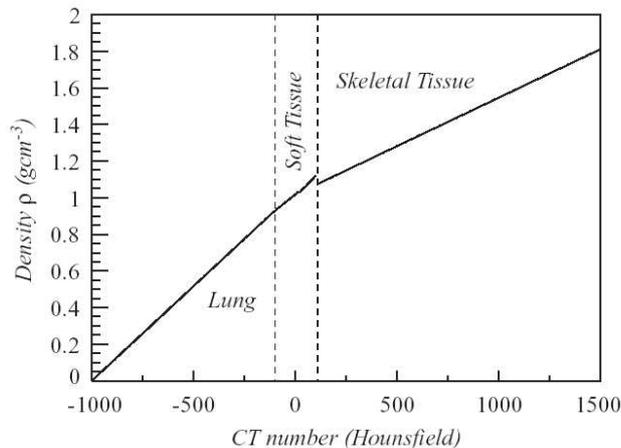


FIG. 3: Courbe de conversion d’une valeur CT (*HU*) en densité ($g.cm^{-3}$). De gauche à droite : les poumons, les tissus mous et l’os.

La méthode de segmentation est la suivante : on seuille les valeurs CT grâce à plusieurs fonctions linéaires de correspondances entre *HU* et densité (en $g.cm^{-3}$), propres au scanner (un exemple est illustré sur la figure 3, issue de [16]). On emploie généralement un seuillage parce que les valeurs CT des voxels suffisent (en première approximation)

²Cette densité correspond à la différence entre le coefficient d’atténuation des rayons X du tissu étudié et le coefficient d’atténuation de l’eau rapportée au coefficient d’atténuation de l’eau. Ainsi, la densité radiologique de l’eau a une valeur de 0 *HU* et celle de l’air de -1000 *HU*.

à indiquer quels sont les matériaux qui y sont représentés. Comme les courbes que l'on manipule dépendent du scanner et sont étalonnées, les densités calculées sont proches de la réalité.

Chaque classe de matériau a une valeur minimale et une valeur maximale d'appartenance (par exemple pour le poumon, entre -950 et -28 *HU*). On calcule ensuite avec la fonction correspondante la densité exacte du matériau représenté par l'intensité CT. Dans [16], les auteurs fixent six classes qui représentent l'air extérieur au patient, les poumons, les tissus mous (avec trois variantes) et les os.

2.2 Simulation physique de radiothérapie avec Geant4

Nous pouvons lire dans [20] que plusieurs codes de simulations de Monte Carlo sont utilisés pour simuler les traitements par radiothérapie. On peut citer MCNPX, EGS4, ou encore BEAM. Le logiciel qui a été choisi dans l'équipe est Geant4, qui est en fait une bibliothèque C++ pour la simulation de hautes énergies. Elle a été validée en astrophysique [14], pour la physique hadronique [6], et plus généralement en physique médicale [7]. On peut lire également un aperçu complet de l'application de Geant4 dans la physique médicale dans [5]. Des chercheurs du CLB travaillent également sur l'utilisation de Geant4 en hadronthérapie.

Les travaux de H. Paganetti sont actuellement une référence dans la simulation de radiothérapie. Le modèle 3D qu'il propose dans [12] grâce aux classes de Geant4 s'appuie sur une matrice de voxels. Il propose une construction géométrique et des optimisations de code que nous détaillerons dans la partie 3.2. Dans [10], on peut lire l'emploi de Geant4 pour la simulation 4D de la tête d'un accélérateur pour l'IMRT (*Intensity Modulated Radiation Therapy*), et dans [15] des détails sur le modèle 4D de Paganetti basé sur Geant4.

Les données du patient que nous voulons intégrer dans le simulateur sont obtenues à partir d'une acquisition scanner. La structure géométrique de l'image 3D est une matrice de voxels. La recherche naïve d'un élément de la matrice (pour connaître l'emplacement de la particule notamment) est en $O(n^3)$, où n est le nombre total de voxels. Dans [12], les auteurs proposent d'utiliser les classes de Geant4 afin de décrire au mieux l'image 3D. Ils se basent en premier lieu sur une découpe en tranches (*smart voxels*) pour hiérarchiser l'image : ces *smart voxels* sont les descendants directs de la racine (monde) et contiennent à leur tour les voxels. La recherche d'un voxel possède une complexité en $O(n^2)$. Grâce à leurs améliorations dans Geant4, les auteurs affirment gagner 8 heures sur une simulation, en considérant qu'avec un cluster de 30 PC classiques, une image de 38 millions de voxels, 600 millions de particules, il faut 30 heures de simulation au total.

Dans cette étude, la construction géométrique que nous décrivons dans la partie 3.2 se base sur les mêmes principes de modélisation que ces travaux. La seule différence étant la simulation de photons à la place de protons. Les processus physiques entrant en jeu étant différents entre ces deux types de particules, nous ne pourrions comparer nos temps de calcul aux leurs. La prise en compte d'une grille irrégulière simple et l'amélioration du code proposé grâce à la recherche d'un voxel en temps constant (partie 3.3) nous ont permis d'accélérer sensiblement la simulation.

3 Représentation d’images 3D dans une simulation physique de Monte Carlo

3.1 Segmentation

Avant de segmenter les images, des algorithmes de filtrage gaussien (filtrage récursif de Deriche [9]) et de rééchantillonnage sont utilisés pour atteindre un résultat approuvé par les médecins médicaux avec lesquels nous avons travaillé; faute de place nous ne détaillerons pas le processus complet dans ce rapport.

Il faut remarquer que la suite de traitements que nous proposons doit être étudiée attentivement. En effet, si l’on interpole lors du rééchantillonnage, de nouvelles intensités vont apparaître dans l’image. A l’emplacement de ces valeurs, des classes intermédiaires seront attribuées alors qu’elles n’existaient pas dans l’image originale. Le traitement d’une image CT dans le but de la segmenter est une étude importante qui sort du cadre de ce rapport. Il serait toutefois intéressant d’approfondir cet axe de recherche.

Rappelons que la segmentation de l’image permet de réduire le nombre de valeurs de 3000 à une dizaine au plus (voir 2.1). Nous avons décidé de consulter des médecins pour connaître les matériaux nécessaires à une simulation pertinente et pour déterminer leur densité. Cette densité n’est pas calculée suivant la valeur CT (comme dans [16]), mais fixée suivant le matériau considéré. Comme les matériaux sont calculés avant la simulation, nous avons opté pour cette solution simple. Cela implique qu’il existe quatre densités différentes dans le scan, si l’on applique la table de correspondances 1.

Intervalle (en HU)	Nom	Densité (en g.cm^{-3})
$[-1000, -950]$	Air	0.0012
$[-950, -700]$	Poumon	0.30
$[-700, 125]$	Tissus mous	1.2
$[125, 2000]$	Os	2.1

TAB. 1: Un exemple de correspondance entre unités Hounsfield et matériau.

Dans la figure 4 (c), nous segmentons l’image (a) suivant le tableau 1. Un voxel de l’image finale est codé sur 1 octet, au lieu de 2 octets, codage initial des examens CT.

Voyons maintenant comment employer cette image segmentée dans une simulation de Monte Carlo.

3.2 Simulations sur une grille 3D régulière

3.2.1 Processus de simulation dans Geant4

Comme nous l’avons vu précédemment, une simulation de Monte Carlo est d’abord décrite par des paramètres géométriques et physiques, mais elle repose aussi sur un paramètre “informatique” important : le nombre de particules à simuler. Le processus

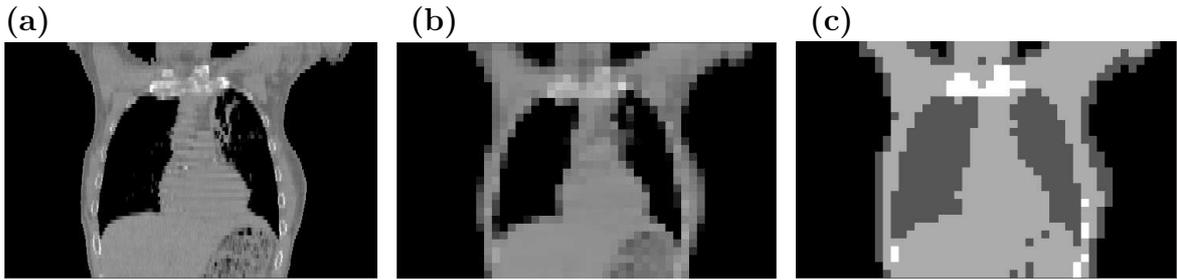


FIG. 4: Processus de segmentation d’une image 3D. (a) : l’image originale (coupe coronale) en $0.94 \times 0.94 \times 5 \text{ mm}^3$, soit 7 millions de voxels environ. (b) : rééchantillonnage de (a) en $10 \times 10 \times 10 \text{ mm}^3$ (30 000 voxels), volontairement grossier pour plus de clarté. (c) : segmentation en quatre parties de (b).

global procède par simulation de chaque particule dite “source”. Plus le nombre de ces particules augmente, plus la simulation s’approche de la réalité. Par contre, les temps de calcul sont importants. En pratique, de un à plusieurs millions de particules sont nécessaires suivant la complexité de la scène et de la précision du calcul de dose voulue.

L’algorithme du processus de simulation est détaillé dans [2, 23, 4], où l’on décrit aussi les classes, les fonctions et les méthodes impliquées dans ce processus. Pour résumer le fonctionnement de **Geant4**, on peut dire qu’une simulation est divisée en plusieurs couches.

Classe	Description
<i>run</i>	Exécution principale de la simulation
<i>event</i>	Création d’une particule à partir de la source. Chaque <i>event</i> peut présenter des caractéristiques différentes (position, direction, énergie, etc.)
<i>track</i>	Etat de la particule à un temps t (vitesse, masse, énergie, etc.). Un <i>track</i> est effacé quand il sort du monde ou quand son énergie cinétique tombe à zéro par exemple
<i>step</i>	Simulation pas à pas de la particule sur un temps dt .
<i>hit</i>	Interaction (<i>step</i> où il y a dépôt d’énergie) dans un volume “sensitif”. La dose est relevée dans chacun de ces volumes sensibles, à la fin de la simulation.

TAB. 2: Principales classes de **Geant4**.

Lorsque l’utilisateur lance une simulation, un *run* est exécuté. Ce processus instancie toutes les classes singletons qui gèrent les principales étapes de l’algorithme. Nous avons résumé les éléments principaux de **Geant4** dans le tableau 2, ainsi que l’enchaînement de leur intervention dans la figure 5.

Geant4 est une bibliothèque complexe qui regroupe au total près de deux milles classes écrites sur un million de lignes de code en **C++**. Nous allons maintenant décrire comment modéliser et manipuler une matrice de voxels en trois dimensions grâce à **Geant4**.

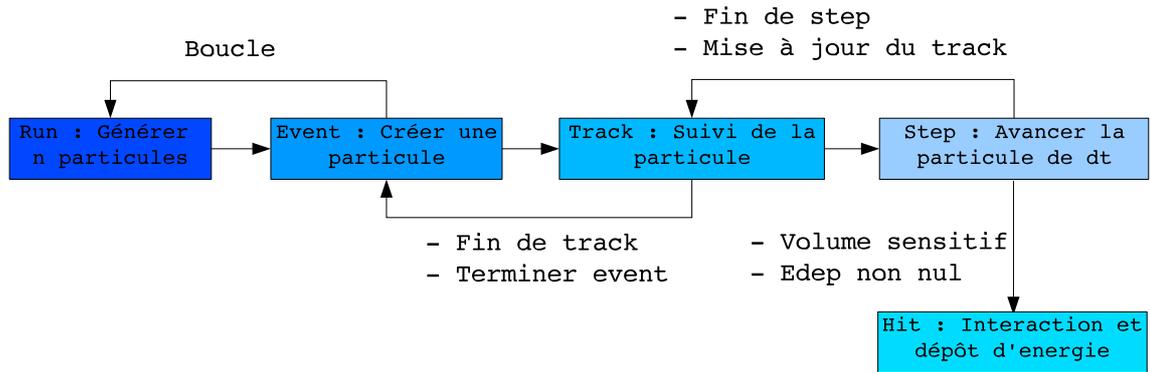


FIG. 5: Synoptique d’une simulation sous Geant4.

3.2.2 Modélisation géométrique

Le patient est représenté par une matrice 3D de voxels et le but est d’obtenir des informations sur la simulation dans chacun d’entre eux (*hits*, *edep*, etc.). Le premier objectif de mon stage a été de représenter l’image 3D dans **Geant4**.

La manière la plus simple est bien sûr de représenter chaque voxel sous forme de parallélépipède (une “boîte” sous **Geant4**), ajouté à la scène par construction CSG (*Constructive Solid Geometry*). Un solide est constitué d’éléments simples (cubes, sphères, etc.) par opérations booléennes (union, intersection, etc.). Le principe est le suivant :

1. On construit un solide S représentant la forme de l’objet, ici une boîte. Seules les dimensions sont renseignées.
2. A partir de S , on instancie un volume logique V_L , qui aura la forme du solide S . On précise ici la composition de V_L : air, eau, fer, graisse, etc.
3. Enfin, on crée un volume physique V_P basé sur V_L . On donne la position du volume.

Par conséquent, la construction d’un voxel implique l’allocation de trois objets. Si l’on considère que tous les voxels ont les mêmes dimensions, il suffit d’allouer un unique objet S pour tous les voxels. Ainsi, si l’on doit coder n voxels, la complexité mémoire est donc en

$$|S| + n \times |V_L| + n \times |V_P| = |S| + (|V_L| + |V_P|)n \quad (1)$$

où $|X|$ représente l’occupation mémoire de X .

Nous avons calculé exactement l’occupation mémoire en octets de chaque objet :

$$|S| = 48 ; |V_L| = 96 ; |V_P| = 64 \quad (2)$$

Avec 15 millions de voxels, on occupe finalement plus de 2 *Go* de la mémoire.

Une première manière de réduire cet espace mémoire est de modifier la résolution des voxels. Un rééchantillonnage de $0.94 \times 5 \times 0.94 \text{ mm}^3$ à $5 \times 5 \times 5 \text{ mm}^3$ permet de réduire sensiblement leur nombre (de $\frac{1}{30}$, soit 500 000 voxels environ), mais on perd évidemment la précision qu’apporte initialement l’image ainsi décrite.

Dans cette étude, nous supposons que la résolution de l'image est décidée suivant la précision de la mesure de l'edep au sein du patient. En clair, si l'on désire un calcul très précis du dépôt de dose dans les voxels, on optera pour une résolution de $1 \times 1 \times 1 \text{ mm}^3$. Au contraire, pour une simulation grossière, ou le calcul du dépôt de dose dans un imageur portal, on pourra choisir $10 \times 10 \times 10 \text{ mm}^3$ par exemple (figure 4 (b)).

Comme la construction CSG classique dans **Geant4** demande des ressources mémoires importantes, cette bibliothèque propose l'emploi de volumes paramétrés. Ces objets requièrent la création d'un seul volume de base décrit par un solide et un volume logique (voir plus haut). Pendant la simulation, lorsque l'on veut connaître un voxel à un indice i donné, le navigateur translate et change le matériau de ce volume de base ³. Ainsi, il n'est plus nécessaire de stocker tous les voxels en mémoire.

Il faut finalement remarquer que les volumes paramétrés demandent des appels de fonctions supplémentaires lors de la recherche d'un volume, alors que l'accès est direct dans une modélisation classique de la matrice. Néanmoins, cette recherche reste rapide, car elle est supportée par des *smart voxels*, ce qui en allège la complexité. Nous nous proposons maintenant d'améliorer cette complexité lors de l'emploi d'une matrice 3D.

3.2.3 Navigation dans la géométrie

Dans **Geant4**, les navigateurs sont des objets permettant de se situer et de se déplacer dans la géométrie décrite par l'utilisateur, quelque soit la méthode de construction (CSG, volumes paramétrés, etc.). Jusqu'à présent, la méthode de localisation d'un volume au sein de la géométrie se base essentiellement sur les *smart voxels*. Ils permettent de diviser virtuellement le patient en tranches (limitées à 1000). Si la particule p est à une position donnée, on recherche d'abord dans quel *smart voxel* SV se situe p , puis on parcourt tous les volumes inclus dans SV pour connaître celui qui contient p . Le principe est appliqué également lorsque l'on veut savoir dans quel voxel la particule poursuit sa trajectoire au *step* suivant.

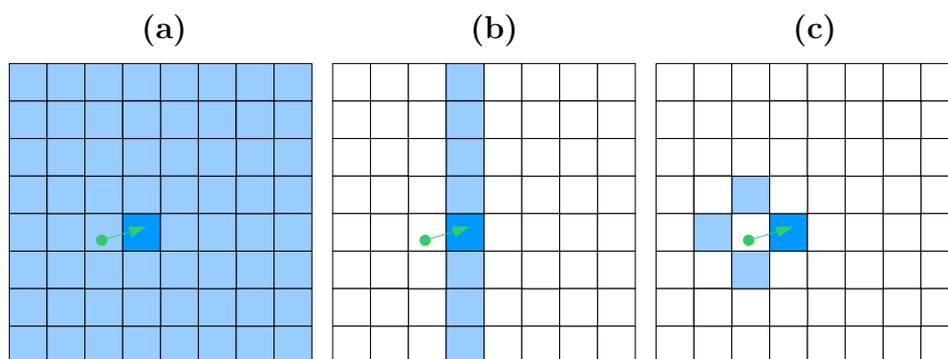


FIG. 6: Recherche du prochain voxel visité par une particule, les voxels testés au pire cas sont en bleu clair. (a) : sans smart voxels. (b) : avec les smart voxels. (c) : avec notre méthode.

³Lors de l'emploi de grilles irrégulières (voir plus loin), on doit aussi spécifier la taille du volume que l'on recherche.

3.2.4 Notre approche

Cette recherche, malgré une complexité en $O(n^2)$, a été améliorée. En effet, il suffit de considérer que chaque voxel a au plus six voisins, de les parcourir et enfin de sélectionner celui qui contient p . Quand on cherche le prochain voxel visité, on ne visite que six voisins car on convient que la probabilité qu’une particule passe par un coin ou une arête est nulle. Il serait intéressant d’étudier les autres configurations de voisinage (18 et 26 voisins) pour confirmer cette théorie.

Contrairement à [12], nous n’utilisons plus les *smart voxels*, qui alourdissent la recherche d’un voxel. Cela nous permet d’accélérer considérablement les simulations (partie 4.2).

Enfin, il faut remarquer que le calcul du dépôt de dose est simple. Il suffit de calculer l’énergie que libère une particule dans le voxel i pour tracer finalement une carte 3D de dose. En chaque case i de cette carte, on représente l’énergie déposée au voxel i au cours de la simulation.

Il peut sembler étrange que cette recherche dans le voisinage d’une matrice 3D (6, 18 ou 26 voisins) ne soit pas déjà prise en compte dans **Geant4**. Il faut rappeler que les logiciels de simulation ne sont que depuis récemment utilisés pour des applications en physique *médicale*. En effet, la plupart des simulations en physique des particules n’impliquent pas la gestion d’une structure aussi complexe que celle d’un patient décrit par un examen scanner de plusieurs millions de voxels. L’insertion de telles données dans ces simulateurs présente actuellement un enjeu majeur de la physique médicale et la complexité du code de **Geant4** (un million de lignes de code) en fait un challenge informatique important.

L’approche que nous proposons pour modéliser le patient en trois dimensions dans **Geant4** nous permet d’accélérer les simulations, en garantissant un calcul de dose précis. Néanmoins, les données de la matrice que nous manipulons révèlent des redondances. De nombreuses parties de l’image sont homogènes, grâce à la segmentation, et affichent un grand nombre de voxels de même intensité (poumons par exemple).

3.3 Vers une simulation sur une grille 3D irrégulière isothétique

3.3.1 Notion de grille irrégulière isothétique

Maintenant que nous avons maîtrisé la modélisation par une grille de voxels, nous voulons construire une grille irrégulière isothétique, pour ensuite décrire de manière plus compacte le patient dans **Geant4** en regroupant les voxels.

On peut lire dans [8] la définition générale des grilles irrégulières isothétiques, dont les problématiques rejoignent aujourd’hui celles de l’arithmétique d’intervalles [1]. Deux contraintes régissent la construction de ces grilles : chaque voxel V est défini par son centre $(x, y, z) \in \mathbb{R}^3$ et sa taille $(l^X, l^Y, l^Z) \in \mathbb{R}^3$. Les contraintes imposées à ces coordonnées peuvent être plus ou moins restrictives. On peut citer entre autres (voir aussi figure 7) :

- Dans le cas d’une grille discrète régulière classique,

$(x, y, z) \in \mathbb{Z}^3$ et $l^X = l^Y = l^Z = 1$.

- Les scanners permettent d'acquérir des grilles anisotropes suivant un axe. On a alors généralement

$$l^X = l^Y = \lambda, \quad l^Z = \mu \text{ et } (x, y, z) = (\lambda i, \lambda j, \mu k) \text{ où } (i, j, k) \in \mathbb{Z}^3.$$

- Une décomposition par *octree* implique que chaque cellule de niveau k (voir plus loin pour plus de précisions) soit de la forme

$$(x, y, z) = \left(\frac{m}{2^k}, \frac{n}{2^k}, \frac{p}{2^k} \right) \text{ et } l^X = \frac{M}{2^{k-1}}, \quad l^Y = \frac{N}{2^{k-1}}, \quad l^Z = \frac{P}{2^{k-1}}$$

où $m, n, p \in \mathbb{Z}$ et M, N, P représentent les dimensions de la grille.

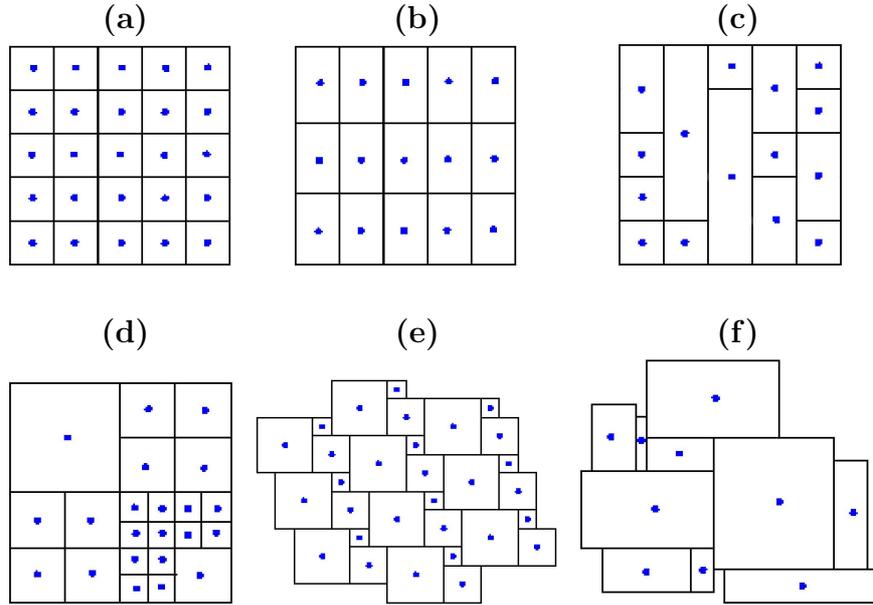


FIG. 7: Exemple de structuration des grilles irrégulières isothétiques. (a) : une grille discrète classique, (b) : une grille anisotrope, (c) : la grille basée sur un RLE que nous proposons d'étudier, (d) : une grille décomposée par un *quadtree*, (e) : une grille où les côtés des plus grosses cellules sont égaux à la somme des côtés des deux autres (f) : une grille irrégulière sans contraintes.

Certaines méthodes de discrétisation irrégulière impliquent qu'une arborescence soit construite en parallèle. Par exemple, le *quad-tree* est une technique où l'on découpe une image 2D (niveau 0) en 4 cellules filles égales en surface (de niveau 1), si celle-ci n'est pas homogène ; le critère d'homogénéité est généralement basé sur les intensités de l'image. On découpe ensuite successivement en 4 chaque cellule de l'image jusqu'à ce que l'image ne soit constituée que de cellules homogènes⁴. Pour connaître les adjacences au sein de l'image finale, on construit un arbre quaternaire pour identifier les liens entre cellules mères et filles (figure 8).

Globalement, on doit souvent étudier le compromis entre le rapport de compression de l'image et la complexité des algorithmes élémentaires comme la recherche d'un voisin. Ici, cette requête requiert le parcours de l'arbre.

⁴En 3D, un *octree* décompose chaque cellule en 8 cellules filles.

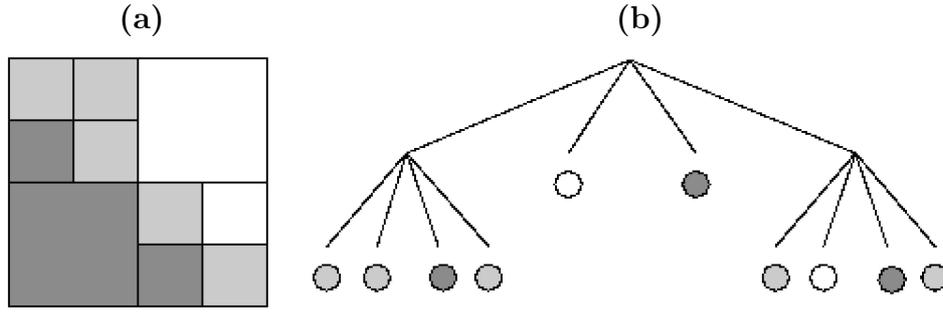


FIG. 8: Une décomposition en quadtree (b) de l'image (a).

3.3.2 Description d'une grille irrégulière isothétique simple

Nous avons choisi de coder notre image 3D grâce à un RLE (*Run Length Encoding*). On parcourt les voxels dans un sens fixé à l'avance (en ligne, en zig-zag) et on les regroupe tant qu'ils ont la même valeur (un exemple est donnée figure 10 (b) et (c)).

Soit G la matrice 3D en entrée, et \overline{G} la grille finalement codée, suivant l'axe des X (il est clair qu'un raisonnement similaire serait présenté pour un autre axe d'encodage). Par analogie à [8], on peut poser $\mathbb{L} \in \mathcal{P}(\mathbb{Z}^3)$ l'ensemble des cellules définies dans \overline{G} . Lors de la construction de notre grille, on cherche à déterminer les fonctions :

$$f_i : \mathbb{Z}^3 \rightarrow \mathbb{L}^3 \quad (3)$$

$$(i, j, k) \rightarrow (x, y = \lambda j, z = \mu k) \quad (4)$$

$$f_t : \mathbb{Z}^3 \rightarrow \mathbb{Z}^3 \quad (5)$$

$$(i, j, k) \rightarrow (l^X, l^Y = \lambda, l^Z = \mu) \quad (6)$$

f_i et f_t permettent de connaître, pour tout indice (i, j, k) de G , la cellule d'indice (x, y, z) dans \overline{G} et sa taille l^X, l^Y, l^Z . Comme le codage est effectué suivant l'axe des X , seule la composante l^X peut changer. On pose donc l^Y et l^Z comme constants. De plus, l'indice (i, j, k) n'est modifié que sur la coordonnée i (qui devient x), les deux autres composantes suivent des règles similaires aux grilles anisotropes.

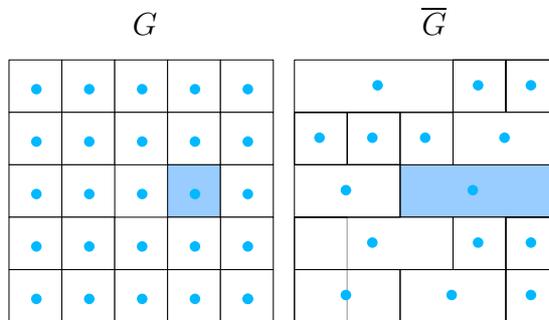


FIG. 9: Mise en correspondance entre G et \overline{G} , où l'on se place à la tranche $k = 1$ par exemple. On a $(i, j, k) = (4, 3, 1)$, $f_i(i, j, k) = (x, j, k) = (2, 3, 1)$ et $f_t(i, j, k) = 3, \lambda, \mu$.

Soit $(i, j, k) \in \mathbb{Z}^3$ tel que $G(i, j, k) \neq G(i - 1, j, k)$ ou $i = 1$, ce qui signifie que l'on commence une nouvelle cellule d'indice (x, j, k) dans la grille \overline{G} . On précise à cette cellule deux informations : la taille $f_t(i, j, k)$ et son indice de départ dans G , (i, j, k) .

On réduit le nombre de volumes élémentaires sensiblement, avec des nuances plus ou moins marquées suivant l'axe dans lequel on code le RLE. Par exemple, pour une image (réelle) initiale de 15 millions de voxels, on obtient environ 3 millions de cellules.

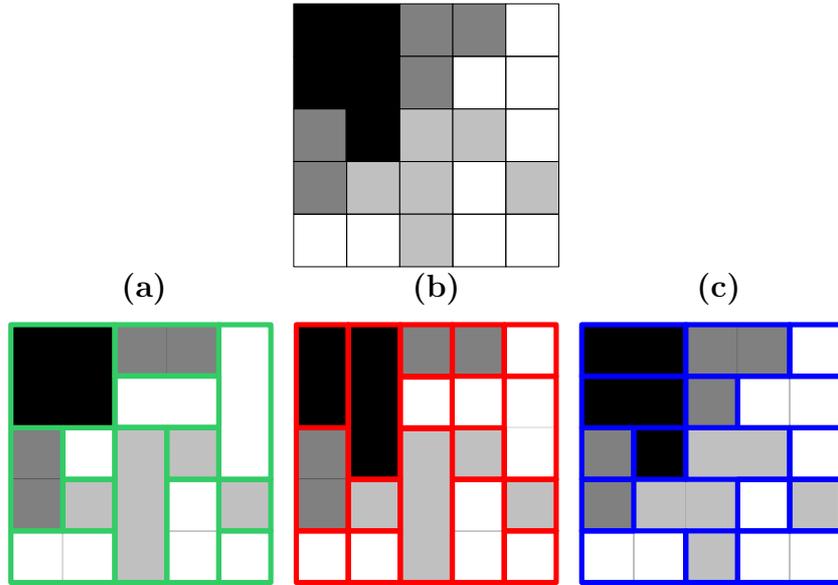


FIG. 10: En haut : une coupe 2D d'une image supportée par une grille régulière. (a) : une grille irrégulière isothétique codant l'image en haut. (b),(c) : encodage RLE suivant l'axe des X et Y respectivement.

Pour décrire le fonctionnement de la navigation dans **Geant4**, nous gardons en mémoire la grille initiale G . Lors de la recherche de la prochaine cellule (x, j, k) visitée par une particule p dans \overline{G} , on se réfère à G pour connaître le voxel (i, j, k) tel que p visite (i, j, k) . On cherche d'abord le voxel qui contient actuellement p , puis on parcourt ses six voisins (comme dans la partie précédente) pour connaître le voxel (i, j, k) vers lequel se dirige la particule. Enfin, on en déduit la cellule de \overline{G} qui contient le voxel (i, j, k) (figure 11) grâce à la fonction f_i .

De même, pour déposer l'énergie d'une particule p , nous considérons la grille G , en cherchant le voxel (i, j, k) où réside p . En effet, le calcul de dose doit rester précis (résolution de G), malgré l'usage de la grille irrégulière.

3.3.3 Bilan algorithmique de notre approche

L'occupation mémoire due à l'emploi de notre grille irrégulière est donc dépendante de la taille de la grille initiale, $|G|$ et de la taille de la grille irrégulière $|\overline{G}|$. Les fonctions f_i et f_t sont définies sous la forme de tables de correspondance entre les deux grilles. Ces tableaux sont de la même taille que G . Notre méthode affiche donc une complexité

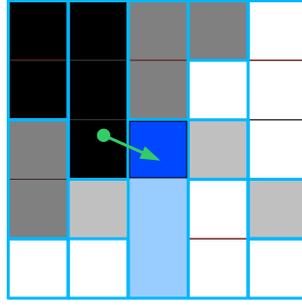


FIG. 11: Recherche de la prochaine cellule (x, j, k) (bleu clair) dans la grille irrégulière \overline{G} . Le voxel recherché (i, j, k) dans G est en bleu foncé.

mémoire en $O(3 \times |G| + |\overline{G}|)$, soit environ 200 *Mo* si l'on code une image de 15 millions de voxels en une grille irrégulière de 500 000 cellules.

Lors de l'accès à une cellule, il faut remarquer que nous faisons encore référence à G pour connaître le voxel sous-jacent cette cellule. Le calcul de dose s'appuie également sur un principe équivalent.

Il est évident qu'une structure de données plus subtile assurerait une complexité mémoire plus faible. Rappelons encore une fois que le but de cette étude est d'accélérer des simulations prenant parfois plusieurs jours, et que la complexité mémoire reste une exigence secondaire.

Une telle structure permet d'optimiser le calcul de dose dans \overline{G} . En effet, le nombre de *hits* est conséquent : pour une simulation de 1 millions d'événements, si l'on dénombre 80 millions de *steps* dans une matrice 3D, on observe généralement 3 à 5 millions de *hits*.

Cependant, rappelons que 90% des 80 millions de *steps* sont bloqués par la géométrie. Le but de notre approche est de diminuer le nombre de *steps* - les particules changent donc moins fréquemment de cellules au sein de la grille.

Dans le cadre du stage, j'ai d'abord implanté la segmentation d'une image issue d'un scan CT en un nombre quelconque de matériaux, définis par un physicien. Puis, j'ai codé les grilles régulières et irrégulières dans **Geant4**, et modifié les classes **C++** originales pour prendre en compte de telles structures. Comme l'emploi de **Geant4** est récent dans l'équipe, il s'agit d'un des premiers travaux basés sur ce logiciel.

Les quelques résultats que nous avons énoncés nous permettent d'introduire la phase de tests que nous avons réalisée afin de quantifier l'efficacité de notre méthode, ainsi que la précision qu'elle affiche lors du calcul de dose.

4 Expérimentations et résultats obtenus

4.1 Matériel et méthode

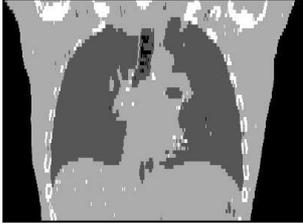
4.1.1 Images de test

Pour comparer les méthodes de construction et de parcours de la géométrie que nous avons décrites jusqu'à présent, nous nous sommes basés sur une image 3D issue d'un scan CT, notée I_0 .

Dans I_0 , nous avons soustrait l'air extérieur au patient, afin d'effectuer nos simulations uniquement sur ce dernier, sans interface supplémentaire. Le nombre de voxels de l'image est diminué environ de moitié. Nous avons ensuite décliné I_0 sous plusieurs résolutions : 1, 2, 5 et 10 mm (résolutions isotropes par simplicité, notons que l'image est sur-échantillonnée suivant l'axe Y pour les résolutions 1 et 2 mm). Ces images I_0 , I_1 , I_2 , I_5 et I_{10} sont ensuite segmentées en quatre matériaux (voir partie 3.1). Dans la table 3, nous récapitulons les tailles de voxels que nous avons choisies, ainsi que le nombre de voxels qui constituent les images de tests.

Nom	Résolutions (en mm^3)	Nombre de voxels
I_0	$0.94 \times 5.0 \times 0.94$	6 614 160
I_1	$1 \times 1 \times 1$	29 059 800
I_2	$2 \times 2 \times 2$	3 623 550
I_5	$5 \times 5 \times 5$	233 280
I_{10}	$10 \times 10 \times 10$	29 520

I_1



I_2



I_5



I_{10}



TAB. 3: Les différentes résolutions d'image pour nos tests. Les images sont rééchantillonnées et segmentées en quatre parties. L'image I_0 n'est pas présentée car très proche de l'image I_1 .

4.1.2 Simulation de référence

La simulation que nous avons choisi de présenter (figure 12) est un cas classique d'irradiation du patient par une source conique, polyénergétique à 6 MeV (*Méga electron Volt*). Cela signifie que chaque particule peut avoir une énergie de départ entre 0 et 6 MeV. La distribution de probabilité qui régit cette énergie initiale est issue des données d'un accélérateur du CLB (voir figure 12). De plus, on peut noter que l'angle d'irradiation est issu de données réelles. Un million d'événements (particules γ) sont générés suivant ces paramètres.

Le patient est placé au centre du monde (dont la composition est de l'air). La source est à 1m au dessus du patient.

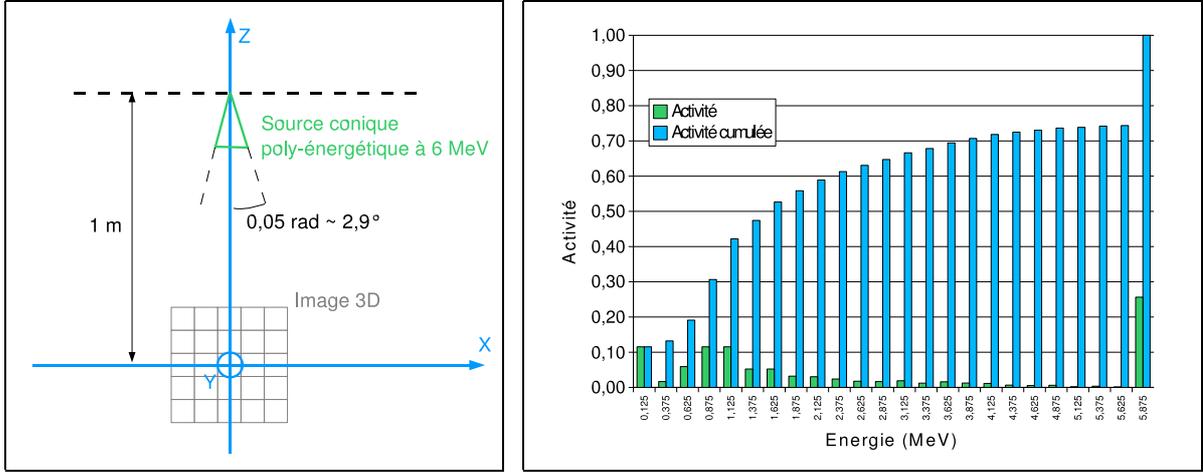


FIG. 12: La simulation de test et le spectre d'énergie de la source.

Pour pouvoir comparer les nouvelles approches détaillées dans ce rapport, nous avons élaboré des cartes de doses de référence, pour toutes les résolutions. Pour ce faire, nous avons exécuté 10 simulations (c'est à dire 10 séquences de nombres aléatoires différentes pour la même simulation) dans la configuration vue plus haut pour chaque image. La méthode employée est celle décrite par H. Paganetti [12] (plus loin notée GR_k , où k représente le sens de construction X , Y ou Z des *smart voxels*).

Les simulations de Monte Carlo sont stochastiques, et dépendent donc d'une séquence de nombres aléatoires. Ainsi, pour déterminer si deux cartes de doses provenant de deux techniques de simulations différentes conduisent à des résultats similaires, il faut que la différence entre ces cartes (dans l'équation 7, membre de gauche) soit inférieure aux erreurs dues aux fluctuations des processus aléatoires (dans l'équation 7, membre de droite).

Ainsi, on calcule la dose moyenne μ_i et la variance σ_i^2 des 10 simulations, pour $i = 0, 1, 2, 5, 10$. Ces cartes vont nous permettre de voir si une simulation de dose totale d_i est correcte :

$$|d_i - \mu_i| < K\sigma_i \quad (7)$$

où K vaut 2 ou 3. Nous avons choisi de poser $K = 3$.

On note pour la suite $M(k)$ la valeur de la matrice M à l'indice $k \in \mathbb{R}^3$.

Si l'on pose N la taille de la matrice de dose, les critères qui nous permettent d'évaluer la qualité d'une simulation de résolution i sont :

- La somme des différences absolues nous permet d'avoir une information directement interprétable.

$$SAD_i = \frac{1}{N} \sum_{k=1, N} |d_i(k) - \mu_i(k)| \quad (8)$$

- la somme des différences au carré, qui évolue de la même manière que le *SAD*, nous permet d’observer et de manipuler ultérieurement un critère quadratique facilement dérivable.

$$SSD_i = \frac{1}{N} \sqrt{\sum_{k=1,N} (d_i(k) - \mu_i(k))^2} \quad (9)$$

- le nombre de voxels cohérents par rapport à l’équation 7 de la carte d_i indique la manière dont cette carte s’éloigne de notre référence par rapport à l’erreur due à la génération de nombres aléatoires.

$$CNT_i = \frac{1}{N} \# \left\{ d_i(k), k = 1, N / |d_i(k) - \mu_i(k)| < K \sigma_i(k) \right\} \quad (10)$$

- le coefficient de corrélation croisé normalisé est un rapport robuste aux légères variations d’intensités, ce qui en fait un critère fiable souvent utilisé.

$$NCC_i = \frac{\sum_{k=1,N} d_i(k) \mu_i(k)}{\sum_{k=1,N} d_i(k)^2 \sum_{k=1,N} \mu_i(k)^2} \quad (11)$$

Les critères que nous avons décrits nous permettent d’évaluer les cartes de dose que nous calculons globalement (*NCC* et *CNT*) ou localement (*SAD* et *SSD*). On peut remarquer, pour résumer que le *SAD* et le *SSD* sont des mesures positives, d’autant plus grandes que la différence avec la référence est importante (différence absolue ou au carré). En parallèle, le *CNT* et le *NCC* sont deux critères compris entre 0 et 1, qui s’approchent de 1 lorsque la carte de dose d est proche de μ (centre de référence).

4.1.3 Méthodes testées

Nous allons tester les méthodes récapitulées dans la table 4. Les techniques se basant sur les grilles régulières sont d’abord issues de l’article de H. Paganetti vu dans l’état de l’art (partie 2.2), puis de notre approche. Les méthodes GI_I^J servent seulement à mettre en avant la méthode GI_+^J que nous avons finalement présentée, sans l’emploi des *smart voxels* fournis par **Geant4**.

Pour tous les tests, nous avons employé les volumes paramétrés. Il est clair qu’une construction classique n’est pas envisageable, même si l’on réduit le nombre de volumes par un RLE (partie 3.2.2).

Nom	Description	Référence
GR_I	Grille régulière et <i>smart voxels</i> dans l’axe I	[12]
GR_+	Grille régulière sans <i>smart voxels</i>	Partie 3.2
GI_I^J	Grille irrégulière d’axe J et <i>smart voxels</i> dans l’axe I	NA
GI_+^J	Grille irrégulière d’axe J sans <i>smart voxels</i>	Partie 3.3

TAB. 4: Les méthodes testées.

Pour illustrer le taux de compression de l’algorithme RLE, nous avons résumé dans la table 5 la taille des grilles irrégulières suivant l’axe d’encodage et la résolution de l’image.

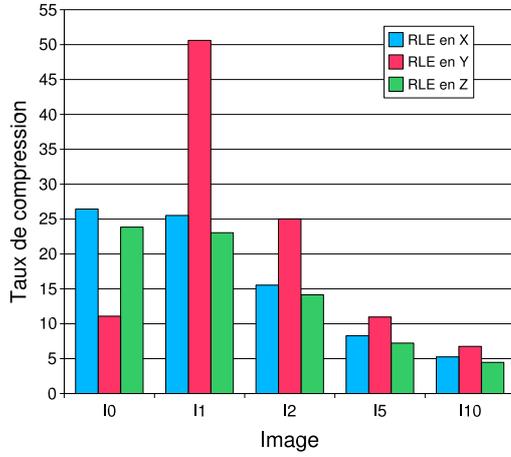


Image	Taille	Axe	Nombre de cellules	Taux
I_0	434	X	250 130	26.44
	60	Y	596 219	11.09
	254	Z	277 375	23.85
I_1	407	X	1 138 603	25.52
	300	Y	574 240	50.61
	238	Z	1 260 516	23.05
I_2	203	X	233 100	15.54
	150	Y	144 707	25.04
	119	Z	256 088	14.15
I_5	81	X	28 165	8.28
	60	Y	21 255	10.97
	48	Z	32 261	7.23
I_{10}	41	X	5 613	5.26
	30	Y	4 367	6.76
	24	Z	6 599	4.47

TAB. 5: Taux de compression grâce à l’algorithme RLE, suivant son axe.

On peut noter que ce ratio dépend de la segmentation que l’on a effectuée au préalable. Si l’on augmente le nombre de classes de matériaux, le taux de compression diminue.

Il faut principalement remarquer dans cette table que le taux de compression ne dépend pas uniquement des dimensions de l’image, mais aussi de son homogénéité. Les images rééchantillonnées sont plus petites si on les code suivant l’axe Y , alors que le plan le plus petit est $Y \times Z$. En effet, si on code dans l’axe des X , comme le plan $Y \times Z$ possède une surface faible, le nombre de lignes de cellules est moindre, par conséquent, le taux de compression devrait être le meilleur. Cela se vérifie dans l’image originale I_0 , qui affiche un taux de compression élevé dans cet axe.

Cela signifie qu’en pratique, pour diminuer au maximum le nombre de cellules, il faut tester tous les axes, puisque la compression dépend des intensités représentées dans l’image 3D. Néanmoins, nous ne cherchons pas à compresser l’image au maximum, mais à garantir un équilibre entre taux de compression et codage efficace de la relation d’adjacence dans notre structure, pour assurer des simulations rapides et précises.

4.2 Résultats

Pour chaque simulation, nous donnons les résultats suivants :

- Le temps total de la simulation, à titre indicatif, sur un PC 2.8 Ghz avec 2 Go RAM, T (en minutes ’ ou heures h) ;
- le débit de simulation D (en particules par seconde) ;
- le nombre de *steps* total S (en millions) ;
- la proportion de *steps* aboutissant à un processus physique SP par rapport à S (en %) ;

- la proportion de *steps* bloqués par la géométrie SG par rapport à S (en %);
- les critères de qualité évoqués plus haut.

4.2.1 Observations générales

Voyons tout d’abord les résultats sur l’ensemble des tests, que l’on a récapitulés en annexe, page 31.

Vitesse d’exécution On peut remarquer que notre choix de ne pas utiliser les *smart voxels* accélèrent grandement les simulations sur une grille régulière (environ deux fois plus rapide selon les résolutions), avec une qualité de calcul de dose très proche de la méthode de référence GR_I ; tous les critères sont du même ordre, avec ou sans les *smart voxels*. Cela paraît logique, puisque le mode de calcul de dose est le même (accès direct au voxel où l’on souhaite calculer l’énergie déposée par la particule).

Les méthodes GI_J^J semblent *a priori* afficher des résultats contradictoires. Lorsque $J \neq I$, les temps concurrencent largement la méthode de H. Paganetti, et même la méthode GR_+ si $J = Z$. Dans ce cas, on peut penser que comme les cellules sont orientées dans le sens de l’irradiation, les particules traversent plus vite le patient (le nombre total de *steps* diminue de 35% en 10 *mm* et de 50% en 2 *mm*).

Le fait que les simulations soient longues quand $I = J$ peut s’expliquer. Dans ce cas, on code les cellules le long de J , et donc elles ont des dimensions inchangées dans les autres axes. En parallèle, on ajoute les *smart voxels*, qui coupent le patient en tranches suivant ce même axe J . Finalement, l’avantage d’avoir des cellules de grande taille dans cette direction est perdu : on a construit virtuellement une grille régulière. Par conséquent, les temps induits par les méthodes GI_J^J sont équivalents à ceux calculés pour GR_J .

Enfin, Notre approche GI_+^J est de loin la plus rapide, avec des temps équivalents à la méthode GR_+ lorsque l’on code dans les axes X et Y , et une réduction de l’ordre de 50% lorsque l’on choisit l’axe Z . Pourtant, on peut remarquer que l’encodage RLE dans cet axe est le moins performant (voir tableau 5).

Précision Les critères SAD et SSD montrent que la dose est localement différente selon la méthode choisie. En effet, on remarque en premier lieu qu’ils sont constants dans le cas des grilles régulières avec ou sans *smart voxels*, et lorsque l’on encode le RLE suivant X ou Y . Par contre, on observe que ces coefficients augmentent de 30 à 40 % si l’on choisit l’axe des Z pour construire notre grille irrégulière. Puisque l’irradiation s’effectue dans cet axe, on peut supposer que les particules déposent leur énergie à des endroits différents dans la carte de dose.

Le CNT est un critère constant, quelle que soit la méthode choisie pour une image donnée. Cette valeur globale indique que le dépôt de dose respecte les fluctuations des processus aléatoires $K\sigma$, même dans une grille irrégulière. De plus, ce critère diminue lorsque la résolution de l’image augmente (0.29 en 1 *mm* contre 0.90 en 10 *mm*); cela signifie que la proportion de voxels qui vérifient l’équation 7 baisse. Une étude plus poussée sur l’emplacement de ces voxels nous indiquerait quelles régions de l’image restent correctes vis à vis de cette équation.

Enfin, on observe que le NCC est fort dans toutes les résolutions si l'on choisit les grilles régulières (plus de 90 %) ; il est le même si l'on code les *smart voxels* ou pas. Il en est de même si l'on code le RLE dans l'axe X ou Y (moins de 1 % d'écart). Dans toutes ces configurations, la dose reste globalement la même. Si l'on code la grille irrégulière suivant Z , on observe une diminution du NCC d'autant plus forte que la résolution augmente : de 0.1 % en 10 mm à 8 % en 1 mm . Cela signifie que le dépôt de dose total est différent avec cette structure.

Il est clair que nous montrons ici une étude relative entre chaque méthode, et qu'une validation expérimentale approuverait l'une d'entre elles. De plus, la segmentation initiale manque de finesse, et un nombre de classes plus grand permettrait certainement de réduire le nombre de volumes dans la géométrie, tout en gardant une certaine précision dans les densités représentées au sein du patient.

4.2.2 Comparaison détaillée

Nous allons maintenant détailler les observations précédentes sur 4 méthodes.

Dans [12], les auteurs choisissent plus précisément l'axe Z pour coder les *smart voxels*, puisque l'irradiation s'effectue comme nous dans cet axe. Nous allons maintenant noter cette technique GR , pour alléger l'écriture. Nous pourrions la comparer avec entre autres notre approche de grilles régulières, $GR+$.

Pour garder une certaine cohérence dans nos observations, nous choisissons de coder les *smart voxels* dans l'axe des Z pour la méthode GI_1^J , et l'axe d'encodage du RLE suivant X (les observations seraient similaires si l'on choisissait Y). On note donc $GI = GI_Z^X$. Enfin, notre dernière contribution dans les grilles irrégulières $GI+ = GI_+^Z$ sera elle aussi comparée avec ces méthodes. Comme l'encodage dans les axes X et Y n'apportent rien par rapport à la méthode $GR+$ dans ce cas, nous ne considérerons que GI_+^Z .

Vitesse d'exécution Les temps de calculs affichés par les différentes méthodes testées sont récapitulés dans la figure 13 en haut. On remarquera d'abord que nous avons classé les images par ordre croissant de leur taille (en nombre de voxels). Les courbes $T = f(I)$ sont strictement croissantes, avec cet ordonnancement ; cela signifie que le temps de simulation dépend de la résolution de l'image. Il est clair que nos approches permettent d'accélérer sensiblement les simulations, quelle que soit la résolution fixée au départ. La méthode de H. Paganetti dénote une complexité de calcul qui semble exploser d'autant que l'image tend vers une résolution précise. Grâce à notre optimisation sur les grilles régulières ($GR+$), cette complexité augmente raisonnablement avec finalement 2 heures de simulation contre 12 heures avec la méthode GR pour l'image I_1 (30 millions de voxels). Les grilles irrégulières permettent de réduire encore les temps de calculs, et notre dernière contribution $GI+$ s'exécute en moins de 30 minutes pour la même image.

Dans la figure 13, on peut voir que le nombre de *steps* diminue grâce aux contributions que nous avons apportées. Plus précisément, le nombre de *steps* qui aboutissent à un processus physique reste presque constant, tandis que l'ensemble de ceux qui sont bloqués par la géométrie réduit grâce aux grilles irrégulières (méthode GI) en premier lieu, puis à notre optimisation ($GI+$). Les particules sont moins souvent arrêtées par la géométrie.

Précision Comme nous l’avons déjà remarqué, la qualité de la dose calculée est globalement la même, quelque soit la méthode employée, mais elle est moins précise localement (voir figure 13 à droite). En effet, on peut supposer que comme la géométrie n’est pas la même avec les méthodes *GI* et *GI+*, le dépôt d’énergie ne s’effectue pas au même endroit dans le patient (les critères *SAD* et *SSD* augmentent), alors que l’ensemble de la carte de dose reste équivalent (les critères *NNC* et *CNT* ne bougent presque pas).

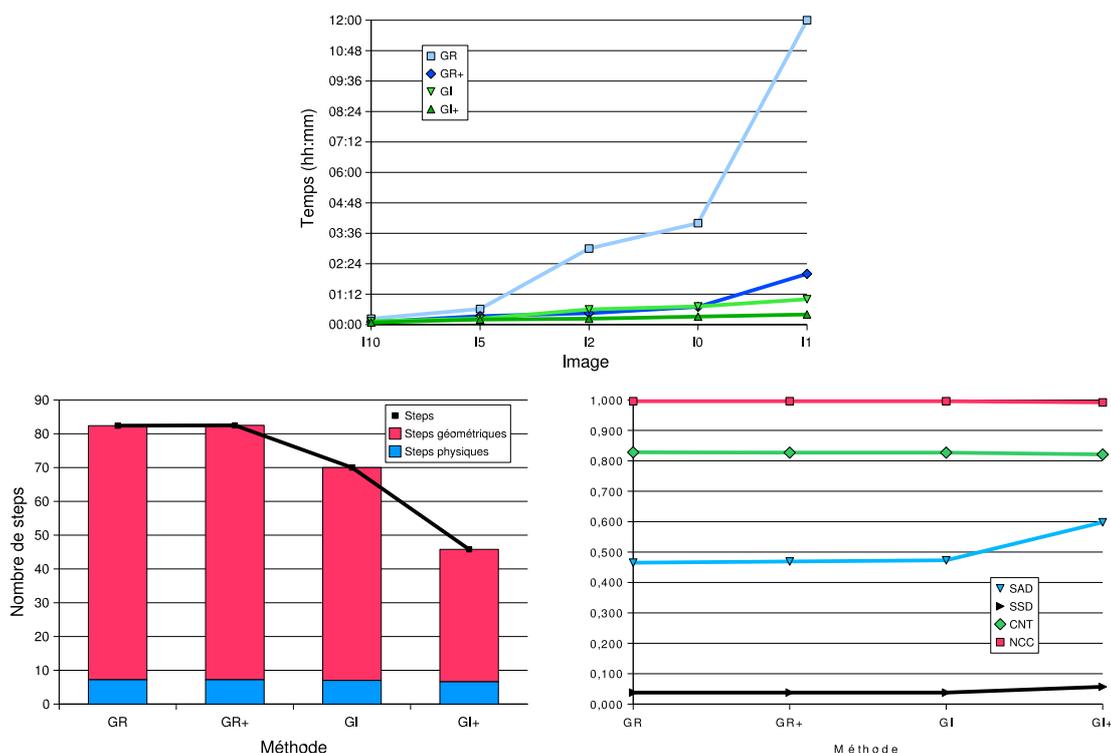


FIG. 13: En haut, les temps de simulations en fonction de l’image testée, pour toutes les approches présentées. A gauche, le nombre de *steps* global, ainsi que les nombres de *steps* physiques et géométriques, en fonction de la méthode employée sur l’image I_5 . A droite, les critères suivant la méthode pour la même image.

Pour exprimer de manière plus fine les différences entre les cartes calculées par chaque méthode, on peut se référer aux doses transmises réelles, lors d’une séance de radiothérapie. Par exemple, lors du traitement du cancer de la prostate, on envoie 60 *Gray* (*Joules/kg*) sur le patient, étalés sur plusieurs séances. En fait, on délivre moins d’une dizaine de *Gray* par séance. Les différences relatives dans les tableaux en annexe A page 31 sont en *MeV*. Pour illustrer au mieux la précision relative d’une méthode, on calcule la différence en *Gray* pour chaque matériau codé (air, poumon, graisse et os).

Pour ce faire, on calcule la différence en *Gray* (*Gy*) pour tout voxel i , $E_G(i)$, grâce à son écart en *MeV*, la densité du matériau représenté en ce voxel (D_{air} , D_{poumon} , $D_{graisse}$, ou D_{os}) et son volume V . Par exemple, la densité vaut $0.00125 \text{ g.cm}^{-3}$ pour l’air, et le voxel a un volume de $2 \times 2 \times 2 \text{ mm}^3 = 8.10^{-3} \text{ cm}^3$.

Ensuite, on ajoute $E_G(i)$ à l’écart global sur le matériau concerné E_{air} , E_{poumon} ,

$E_{graisse}$ ou E_{os} . Ce qui signifie que l'écart sur l'air est

$$E_{air} = \sum_{i=1,N / D=D_{air}} E_G(i) = \sum_{i=1,N / D=D_{air}} \frac{|d(i) - \mu(i)| \times 1.6 \times 10^{-19}}{V \times D} \quad (12)$$

Les différences moyennes sur chaque matériau sont récapitulées dans les tableaux en annexe B, page 33.

Il est difficile de comparer ces valeurs entre chaque image testée. En effet, les différences entre la simulation test et le centre de référence μ sont ici exagérées par l'équation 12. On peut seulement remarquer que ces écarts sont toujours du même ordre (de l'ordre du millionième de *centiGray* ou *cGy*). En fait, quelle que soit l'approche adoptée, on est sûr de la valeur de la dose transmise au patient, avec un écart très faible (comparativement à la dizaine de *Gray* que l'on délivre en une séance).

Pour une image donnée, il faut noter que les écarts sont équivalents pour les méthodes basés sur les grilles régulières et pour la méthode *GI*. Par contre, ils augmentent pour toutes les classes lorsque l'on choisit *GI+* (entre 20 et 30 % de hausse). Une telle différence s'explique encore par le déplacement du dépôt de dose quand la grille est codée par un RLE d'axe *Z*. Entre chaque matériau, l'écart observé ne dépend pas de la proportion des classes choisies (voir tableau 12, en annexe B), mais de leur densité. Plus le matériau est dense, et plus l'écart diminue. Cela s'explique par le fait que les particules sont davantage freinées dans un environnement dense, et déposent donc leur énergie dans une région plus restreinte que dans l'air par exemple, où les photons sont plus mobiles.

Synthèse On peut résumer les observations précédentes dans le tableau 6, qui a été élaboré par rapport à l'image la plus précise I_1 , de résolution $1 \times 1 \times 1 \text{ mm}^3$.

Méthode	Temps (h,m)	SG (%)	CNT	Écart de dose (cGy)
<i>GR</i>	12h	97.5	0.285	$4.66 \cdot 10^{-7}$
<i>GR+</i>	2h	97.5	0.286	$4.70 \cdot 10^{-7}$
<i>GI</i>	1h	97	0.286	$4.75 \cdot 10^{-7}$
<i>GI+</i>	24'	94.6	0.284	$5.83 \cdot 10^{-7}$

TAB. 6: Tableau récapitulatif des remarques de cette partie, pour l'image I_1 . On y indique le temps de simulation en heures, minutes, le pourcentage de *steps* géométriques par rapport au nombre total de *steps*, le critère *CNT*, et l'écart de dose en *centiGray* sur les poumons.

Pour clore cette phase d'expérimentations, on peut ajouter qu'une étude avec un HDV (Histogramme Dose Volume) serait nécessaire pour évaluer de manière plus réaliste et plus précise les méthodes que nous avons implantées. Cela permettrait de prendre en compte les zones cibles (tumeurs) et les régions à risques (organes, etc.) du patient, définies par un physicien. La carte de dose nous indiquerait alors quelle est l'approche la plus adaptée à nos simulations physiques de radiothérapie.

Bilan et perspectives

Dans le cadre de la modélisation 4D du patient que l'on cherche à atteindre au CLB, la représentation géométrique au sein d'un logiciel de simulation tel que **Geant4** reste un problème crucial. En effet, le calcul de dose est précis grâce à une simulation des processus physiques très aboutie, mais les temps de calculs sont prohibitifs, comme on a pu le constater dans la partie 4.2. Nous avons proposé différentes approches originales afin de modéliser dans **Geant4** une cible complexe représentant le patient initialement donné sous la forme d'un scan CT au format DICOM.

Nous avons tout d'abord amélioré la méthode issue des travaux de H. Paganetti, qui se base sur une matrice régulière. Ne pas coder les *smart voxels* nous a permis de réduire les temps de simulations approximativement de moitié, sans modifier le calcul de dose.

L'emploi d'une grille irrégulière simple codée avec un RLE réduit encore les temps de calcul. On divise par 3 ou 4 les temps de simulations par rapport à l'approche de H. Paganetti (par exemple en 5 mm, 1371 parts/sec contre 445 parts/sec). Avec une résolution très précise fixée par exemple à $1 \times 1 \times 1 \text{ mm}^3$, et un million de particules générées, une simulation s'exécute en moins de 30 minutes au lieu de 12 heures. Par contre, on remarque des écarts relatifs locaux de 30 à 40 % entre les cartes de doses en sortie de nos simulation, et les cartes fixées grâce à une grille régulière. Cependant, les cartes sont globalement équivalentes. Il faut observer que nous ne savons pas quelle est la méthode de modélisation du patient la plus efficace pour calculer la dose.

À partir de ce constat, on peut envisager plusieurs axes de recherche.

Les cartes de dose 3D peuvent être validées par des expérimentations réalisées par des physiciens du CLB. Nous pourrions connaître alors quelle technique s'approche le plus de la réalité en évaluant une erreur absolue (et non plus une erreur relative, comme nous le faisons dans cette étude).

En parallèle, on peut coder différentes méthodes de subdivision et de construction de grilles irrégulières (octree, etc.). Ces approches peuvent être elles-mêmes approuvées ou rejetées par la validation expérimentale. Il faut rappeler également qu'il faut étudier le compromis entre compression de l'image en entrée et codage de la relation d'adjacence de la structure de données que l'on aura choisie.

À partir de cette validation, si les grilles irrégulières s'avèrent être le support le plus adapté à une simulation correcte, il est clair qu'il faut alors déterminer quelle est la configuration de grille qui optimise à la fois les temps de calculs et par exemple les critères que nous avons cités dans notre phase de tests.

Par contre, si les grilles régulières restent un standard dans les simulations de radiothérapie, on peut envisager la construction d'une matrice hybride, mêlant structure régulière dans les zones de l'image où le calcul de dose doit être très précis (région de la tumeur notamment), et cellules irrégulières dans le reste de l'image. Cette représentation nécessite donc une information *a priori* sur la localisation de la tumeur et les zones de l'image balayées par le faisceau incident. Cette tâche peut aisément s'inscrire dans le système de simulation et de validation des paramètres d'une séance de radiothérapie.

Références

- [1] P. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 23 of *Contemporary Mathematics*, pages 1–56. American Mathematical Society Press, Providence, RI, USA, 1999.
- [2] S. Agostinelli and al. Geant4 - A Simulation Toolkit. *Nuclear Instruments and Methods*, A506(3) :250–303, July 2003.
- [3] A. Amako. A Short Course on Geant4 Simulation Toolkit. In *IEEE Nuclear Science Symposium*, Roma, October 2004.
<http://www.ge.infn.it/geant4/events/nss2004/geant4course.html>.
- [4] A. Amako and al. *Geant4 User's Guide - For Toolkit Developers*.
<http://wwwasd.web.cern.ch/wwwasd/geant4/G4UsersDocuments/UsersGuides/ForToolkitDeveloper/html/index.html>.
- [5] L. Archambault and al. Overview of Geant4 Application in Medical Physics. In *IEEE Nuclear Science Symposium*, Portland, October 2003.
www.ge.infn.it/geant4/papers/2003/nss2003/IEEE2003medical.pdf.
- [6] J. Beringer and al. Validation of Geant4 Hadronic Physics. In *IEEE Nuclear Science Symposium*, Portland, October 2003.
lcgapp.cern.ch/project/simu/validation/docs/NSSPortland-paper.pdf.
- [7] J.-F. Carrier, L. Archambault, and L. Beaulieu. Validation of Geant4, an object-oriented Monte Carlo toolkit for simulation in medical physics. *Medical Physics*, 31(3) :484–492, March 2004.
- [8] D. Coeurjolly. Supercover model and digital straight line recognition on irregular isothetic grids. In E. Andres, G. Damiand, and P. Lienhardt, editors, *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 311–322. Springer, 2005.
- [9] R. Deriche. Recursively implementing the gaussian and its derivatives. Technical Report 1893, INRIA, April 1993.
- [10] F. Foppiano, B. Mascialino, MG. Pia, and M. Piergentili. Geant4 simulation of an accelerator head for intensity modulated radiotherapy. In *The Monte Carlo Method : Versatility Unbounded in a Dynamic Computing World*, Chattanooga, Tennessee, April 2005.
- [11] V. Havran. *Heuristic Ray Shooting Algorithms*. PhD thesis, Faculty of Electrical Engineering, November 2000.
- [12] H. Jiang and H. Paganetti. Adaptation of Geant4 to Monte Carlo dose calculations based on CT data. *Medical Physics*, 31(10) :2811–2818, 17 September 2004.
- [13] B. Ma and R.E. Ellis. Surface-based registration with a particle filter. In Christian Barillot, David R. Haynor, and Pierre Hellier, editors, *MICCAI (1)*, volume 3216 of *Lecture Notes in Computer Science*, pages 566–573. Springer, 2004.
- [14] A. Mantero and al. Simulation of X-Ray Fluorescence and Application to Planetary Astrophysics. Technical report, Geant4 Space Users, 2004.

- [15] H. Paganetti. Four-dimensional Monte Carlo simulation of time-dependant geometries. *Phys. Med. Biol.*, 49(6) :75–81, 2004.
- [16] P. Rodrigues and al. Application of Geant4 radiation transport toolkit to dose calculations in anthropomorphic phantoms. *Applied Radiation and Isotopes*, 61 :1451–1461, 2004.
- [17] W. Schneider, T. Bortfeld, and W. Schlegel. Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions. *Phys. Med. Biol.*, 45 :459–478, 2000.
- [18] A. Vacavant and T. Chateau. Realtime head and hands tracking in monocular vision. In *IEEE International Conference on Image Processing 2005*, 2005. En cours de parution.
- [19] F. Verhaegen and S. Devic. Sensitivity study for CT image use in Monte Carlo treatment planning. *Phys. Med. Biol.*, 50 :937–946, 2005.
- [20] F. Verhaegen and J. Seuntjens. Monte carlo modelling of external radiotherapy photon beams. *Phys. Med. Biol.*, 48(7) :107–164, November 2003.
- [21] Z. Waters. Global Illumination.
http://www.cs.wpi.edu/~emmanuel/ISPs/zack_isp/IndependentStudy.html.
- [22] H-P. Wellisch, M. Maire, and L. Urban. *Geant4 Physics Reference Manual*.
<http://wwwasd.web.cern.ch/wwwasd/geant4/G4UsersDocuments/UsersGuides/PhysicsReferenceManual/html/PhysicsReferenceManual.html>.
- [23] D.H. Wright and al. *Geant4 User's Guide - For Application Developers*.
<http://wwwasd.web.cern.ch/wwwasd/geant4/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/index.html>.

ANNEXE A

Résultats des simulations tests

Méthode	T	D	S	SP	SG	SAD	SSD	CNT	NCC
GR_X	4h	82	343.6	2.67	97.33	0.046	$7.44 \cdot 10^{-5}$	0.549	0.970
GR_Y	3h	102	343.4	2.66	97.34	0.046	$7.43 \cdot 10^{-5}$	0.549	0.970
GR_Z	4h	75	343.3	2.66	97.34	0.046	$7.42 \cdot 10^{-5}$	0.549	0.970
GR_+	42'	400	343.7	2.66	97.34	0.046	$7.45 \cdot 10^{-5}$	0.549	0.970
GI_X^X	2h	123	270.2	3.14	96.86	0.047	$7.69 \cdot 10^{-5}$	0.550	0.968
GI_Y^X	1h	215	270.0	3.15	96.85	0.047	$7.66 \cdot 10^{-5}$	0.549	0.968
GI_Z^X	43'	392	270.2	3.15	96.85	0.047	$7.72 \cdot 10^{-5}$	0.549	0.969
GI_X^Y	55'	305	331.5	2.73	97.27	0.046	$7.48 \cdot 10^{-5}$	0.549	0.970
GI_Y^Y	2h	133	331.5	2.73	97.27	0.047	$7.48 \cdot 10^{-5}$	0.549	0.969
GI_Z^Y	1h	219	331.6	2.73	97.27	0.046	$7.47 \cdot 10^{-5}$	0.549	0.970
GI_X^Z	27'	616	113.8	6.90	93.10	0.063	$1.21 \cdot 10^{-4}$	0.547	0.931
GI_Y^Z	43'	389	113.9	6.90	93.10	0.064	$1.22 \cdot 10^{-4}$	0.547	0.931
GI_Z^Z	3h	107	113.9	6.90	93.10	0.063	$1.22 \cdot 10^{-4}$	0.549	0.931
GI_+^X	38'	444	270.1	3.15	96.85	0.047	$7.66 \cdot 10^{-5}$	0.550	0.969
GI_+^Y	47'	353	331.4	2.73	97.27	0.046	$7.47 \cdot 10^{-5}$	0.549	0.970
GI_+^Z	19'	885	113.8	6.94	93.06	0.063	$1.21 \cdot 10^{-4}$	0.547	0.931

TAB. 7: Résultats sur I_0 .

Méthode	T	D	S	SP	SG	SAD	SSD	CNT	NCC
GR_X	8h	36	376.0	2.51	97.49	0.018	$1.61 \cdot 10^{-5}$	0.286	0.901
GR_Y	18h	15	376.0	2.51	97.49	0.018	$1.61 \cdot 10^{-5}$	0.286	0.901
GR_Z	12h	23	376.4	2.51	97.49	0.018	$1.61 \cdot 10^{-5}$	0.286	0.901
GR_+	2h	177	376.4	2.51	97.49	0.019	$1.61 \cdot 10^{-5}$	0.285	0.901
GI_X^X	7h	37	307.7	2.92	97.08	0.019	$1.65 \cdot 10^{-5}$	0.286	0.895
GI_Y^X	2h	187	307.8	2.92	97.08	0.018	$1.65 \cdot 10^{-5}$	0.286	0.894
GI_Z^X	1h	195	307.7	2.92	97.08	0.019	$1.65 \cdot 10^{-5}$	0.286	0.895
GI_X^Y	1h	281	311.2	2.90	97.10	0.019	$1.65 \cdot 10^{-5}$	0.286	0.897
GI_Y^Y	17h	16	311.2	2.90	97.10	0.019	$1.64 \cdot 10^{-5}$	0.285	0.897
GI_Z^Y	2h	193	311.3	2.90	97.10	0.019	$1.65 \cdot 10^{-5}$	0.286	0.895
GI_X^Z	55'	301	162.4	5.28	94.62	0.023	$2.27 \cdot 10^{-5}$	0.284	0.832
GI_Y^Z	50'	347	162.4	5.28	94.62	0.023	$2.27 \cdot 10^{-5}$	0.284	0.832
GI_Z^Z	12h	22	162.4	5.28	94.62	0.022	$2.26 \cdot 10^{-5}$	0.284	0.833
GI_+^X	44'	383	307.9	2.92	97.08	0.019	$1.65 \cdot 10^{-5}$	0.286	0.897
GI_+^Y	47'	355	311.3	2.89	97.11	0.019	$1.65 \cdot 10^{-5}$	0.286	0.897
GI_+^Z	24'	399	162.6	5.26	94.64	0.022	$2.27 \cdot 10^{-5}$	0.284	0.833

TAB. 8: Résultats sur I_1 .

Méthode	T	D	S	SP	SG	SAD	SSD	CNT	NCC
GR_X	2h	131	192.7	4.36	95.64	0.080	$1.7 \cdot 10^{-4}$	0.581	0.975
GR_Y	5h	59	192.5	4.36	95.64	0.079	$1.7 \cdot 10^{-4}$	0.581	0.974
GR_Z	3h	88	192.9	4.35	95.65	0.080	$1.7 \cdot 10^{-4}$	0.581	0.975
GR_+	27'	619	192.6	4.36	95.64	0.080	$1.7 \cdot 10^{-4}$	0.581	0.975
GI_X^X	3h	104	159.5	5.08	94.92	0.080	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_Y^X	59'	282	159.5	5.02	94.98	0.081	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_Z^X	36'	459	159.2	5.02	94.98	0.080	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_X^Y	30'	549	160.7	4.98	95.02	0.080	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_Y^Y	3h	88	160.7	4.98	95.02	0.081	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_Z^Y	33'	505	160.7	4.98	95.02	0.080	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_X^Z	26'	626	90.7	8.27	91.73	0.099	$2.5 \cdot 10^{-4}$	0.579	0.954
GI_Y^Z	25'	645	90.7	8.27	91.73	0.099	$2.5 \cdot 10^{-4}$	0.579	0.954
GI_Z^Z	4h	74	90.8	8.37	91.63	0.099	$2.5 \cdot 10^{-4}$	0.579	0.954
GI_+^X	24'	682	159.3	5.08	94.92	0.080	$1.8 \cdot 10^{-4}$	0.582	0.974
GI_+^Y	24'	688	160.7	4.98	95.02	0.081	$1.8 \cdot 10^{-4}$	0.581	0.974
GI_+^Z	14'	1157	90.7	8.27	91.73	0.099	$2.5 \cdot 10^{-4}$	0.579	0.954

TAB. 9: Résultats sur I_2 .

Méthode	T	D	S	SP	SG	SAD	SSD	CNT	NCC
GR_X	28'	597	82.4	8.74	91.26	0.468	0.0038	0.827	0.996
GR_Y	52'	320	82.4	8.74	91.26	0.471	0.0039	0.826	0.996
GR_Z	37'	445	82.4	8.74	91.26	0.465	0.0038	0.828	0.996
GR_+	20'	841	82.5	8.73	91.27	0.469	0.0038	0.827	0.996
GI_X^X	30'	546	69.9	9.87	90.13	0.468	0.0038	0.828	0.996
GI_Y^X	17'	1003	70.0	10.0	90.0	0.476	0.0039	0.827	0.996
GI_Z^X	14'	1156	70.0	10.0	90.0	0.473	0.0039	0.827	0.996
GI_X^Y	13'	1272	70.4	9.80	90.20	0.477	0.0038	0.826	0.996
GI_Y^Y	39'	431	70.5	9.93	90.07	0.475	0.0039	0.828	0.996
GI_Z^Y	17'	965	70.5	9.79	90.21	0.472	0.0038	0.828	0.996
GI_X^Z	19'	873	45.8	14.41	85.59	0.593	0.0058	0.822	0.992
GI_Y^Z	18'	905	45.8	14.41	85.59	0.594	0.0057	0.821	0.992
GI_Z^Z	54'	308	45.9	14.38	85.62	0.599	0.0058	0.820	0.992
GI_+^X	11'	1502	69.9	9.87	90.13	0.473	0.0038	0.826	0.996
GI_+^Y	11'	1512	70.4	9.80	90.20	0.473	0.0038	0.828	0.996
GI_+^Z	12'	1371	45.8	14.41	85.59	0.598	0.0057	0.821	0.992

TAB. 10: Résultats sur I_5 .

Méthode	T	D	S	SP	SG	SAD	SSD	CNT	NCC
GR_X	12'	1324	44.9	14.48	85.52	1.629	0.037	0.894	0.999
GR_Y	17'	983	44.9	14.48	85.52	1.633	0.037	0.895	0.999
GR_Z	14'	1173	44.9	14.48	85.52	1.617	0.038	0.894	0.999
GR_+	06'	2481	44.8	14.29	85.61	1.601	0.037	0.894	0.999
GI_X^X	12'	1377	39.2	16.07	83.93	1.661	0.038	0.894	0.999
GI_Y^X	08'	1934	39.2	16.07	83.93	1.596	0.036	0.898	0.999
GI_Z^X	08'	1951	39.2	16.07	83.93	1.633	0.037	0.899	0.999
GI_X^Y	09'	1855	39.4	15.99	84.01	1.689	0.039	0.899	0.999
GI_Y^Y	13'	1241	39.4	15.99	84.01	1.611	0.037	0.895	0.999
GI_Z^Y	09'	1870	39.4	15.99	84.01	1.659	0.039	0.896	0.999
GI_X^Z	06'	2576	29.2	20.89	79.11	2.139	0.059	0.885	0.998
GI_Y^Z	06'	2672	29.2	20.89	79.11	2.184	0.059	0.887	0.998
GI_Z^Z	13'	1245	29.2	20.89	79.11	2.200	0.060	0.883	0.998
GI_+^X	10'	1632	39.2	16.07	83.93	1.653	0.039	0.893	0.999
GI_+^Y	07'	2288	39.4	15.99	84.01	1.659	0.038	0.897	0.999
GI_+^Z	05'	3334	29.2	20.89	79.11	2.156	0.060	0.888	0.998

TAB. 11: Résultats sur I_{10} .

ANNEXE B

Différences observées par matériau

Classe	Effectif
Air	43 %
Poumon	16 %
Graisse	35 %
Os	6 %

TAB. 12: Effectif de chaque classe de matériau pour l'image de test I_0 . Ces effectifs sont valables pour toutes les images construites à partir de I_0 .

Méthode	Air	Poumon	Graisse	Os
GR	$8.35 \cdot 10^{-9}$	$4.68 \cdot 10^{-9}$	$2.61 \cdot 10^{-9}$	$2.73 \cdot 10^{-9}$
GR_+	$8.31 \cdot 10^{-9}$	$4.69 \cdot 10^{-9}$	$2.63 \cdot 10^{-9}$	$2.73 \cdot 10^{-9}$
GI	$8.27 \cdot 10^{-9}$	$4.89 \cdot 10^{-9}$	$2.69 \cdot 10^{-9}$	$2.76 \cdot 10^{-9}$
GI_+	$1.21 \cdot 10^{-8}$	$5.91 \cdot 10^{-9}$	$3.71 \cdot 10^{-9}$	$3.64 \cdot 10^{-9}$

TAB. 13: Différences moyennes sur les quatre matériaux codés (*Gray*) pour l'image I_0 .

Méthode	Air	Poumon	Graisse	Os
<i>GR</i>	$1.19 \cdot 10^{-8}$	$7.29 \cdot 10^{-9}$	$4.66 \cdot 10^{-9}$	$5.04 \cdot 10^{-9}$
<i>GR+</i>	$1.20 \cdot 10^{-8}$	$7.28 \cdot 10^{-9}$	$4.70 \cdot 10^{-9}$	$5.05 \cdot 10^{-9}$
<i>GI</i>	$1.22 \cdot 10^{-8}$	$7.49 \cdot 10^{-9}$	$4.75 \cdot 10^{-9}$	$5.12 \cdot 10^{-9}$
<i>GI+</i>	$1.55 \cdot 10^{-8}$	$8.35 \cdot 10^{-9}$	$5.83 \cdot 10^{-9}$	$6.04 \cdot 10^{-9}$

TAB. 14: Différences moyennes sur les quatre matériaux codés (*Gray*) pour l'image I_1 .

Méthode	Air	Poumon	Graisse	Os
<i>GR</i>	1.75×10^{-8}	4.55×10^{-9}	2.49×10^{-9}	2.55×10^{-9}
<i>GR+</i>	1.73×10^{-8}	4.54×10^{-9}	2.48×10^{-9}	2.55×10^{-9}
<i>GI</i>	1.74×10^{-8}	4.63×10^{-9}	2.51×10^{-9}	2.57×10^{-9}
<i>GI+</i>	2.34×10^{-8}	5.31×10^{-9}	3.15×10^{-9}	3.08×10^{-9}

TAB. 15: Différences moyennes sur les quatre matériaux codés (*Gray*) pour l'image I_2 .

Méthode	Air	Poumon	Graisse	Os
<i>GR</i>	1.86×10^{-8}	1.99×10^{-9}	9.11×10^{-10}	8.61×10^{-10}
<i>GR+</i>	1.92×10^{-8}	2.01×10^{-9}	9.16×10^{-10}	8.73×10^{-10}
<i>GI</i>	1.92×10^{-8}	2.03×10^{-9}	9.24×10^{-10}	8.80×10^{-10}
<i>GI+</i>	2.58×10^{-8}	2.46×10^{-9}	1.19×10^{-9}	1.08×10^{-9}

TAB. 16: Différences moyennes sur les quatre matériaux codés (*Gray*) pour l'image I_5 .

Méthode	Air	Poumon	Graisse	Os
<i>GR</i>	1.35×10^{-8}	9.99×10^{-10}	3.95×10^{-10}	3.34×10^{-10}
<i>GR+</i>	1.44×10^{-8}	1.00×10^{-9}	3.88×10^{-10}	3.31×10^{-10}
<i>GI</i>	1.44×10^{-8}	9.94×10^{-10}	3.97×10^{-10}	3.47×10^{-10}
<i>GI+</i>	1.79×10^{-8}	1.29×10^{-9}	5.33×10^{-10}	4.38×10^{-10}

TAB. 17: Différences moyennes sur les quatre matériaux codés (*Gray*) pour l'image I_{10} .