

TP 5: Reconstruction tomographique

Auteur: Dmitri Matenine et Philippe Després

Date de remise: 23 février 2026

PHY-3500 – Physique numérique (H26)

Professeur : Antoine Allard, Thomas Labbe

Objectif

Effectuer des reconstructions tomographiques à l'aide du langage de programmation Python à partir de données de projection. Se familiariser avec diverses approches classiques de reconstruction tomographique.

Introduction

La *tomographie* est un terme à portée très large qui désigne plusieurs techniques d'imagerie. Le terme provient du mot grec *tomos* qui veut dire tranche. Il s'agit donc de techniques conçues pour imager l'intérieur d'un objet physique tranche par tranche de façon non-invasive/non-destructive. Ce travail pratique se concentre sur une modalité en particulier, la tomodensitométrie (TDM), appelée Computed Tomography (CT) dans la littérature anglophone ou TACO dans la région de Québec (probablement pour tomographie axiale calculée par ordinateur). Il s'agit d'une méthode où l'on effectue plusieurs mesures d'atténuation partielle d'un faisceau de rayons à travers un objet dans le but d'estimer la distribution de la densité de l'objet. Le type de radiation dépend de l'objet à imager. Ainsi, les objets semi-transparents à la lumière visible sont imagés via la tomographie optique. Le corps humain est exploré par des rayons X d'énergies comprises normalement entre 30 keV et 140 keV. Des objets de fabrication industrielle, comme des moteurs de fusée ou des échantillons de minéraux sont imagés par des rayons gamma d'énergies diverses.

L'acquisition de données brutes se fait en tournant un statif composé d'une source de radiation et d'un détecteur autour de l'objet à imager. On acquiert ainsi des centaines de

projections, typiquement sur 360 degrés autour de l'objet. L'estimation de la distribution des densités se fait par reconstruction tomographique, le principal sujet de ce travail pratique.

Les fondements théoriques de la reconstruction tomographique ont été énoncées par le mathématicien Johann Radon en 1917. Il fallut attendre l'arrivée des ordinateurs pour être en mesure de faire les calculs associés à la reconstruction en des temps raisonnables. Le tomodensitomètre découle des travaux indépendants de Sir Godfrey Hounsfield (vers 1971) et d'Allan McLeod Cormack, qui ont obtenu le prix Nobel en médecine en 1979 pour l'invention de la TDM médicale. Sir Hounsfield travaillait chez EMI (Electrical and Musical Industries), la maison de disque originale des Beatles. L'histoire raconte que sans les revenus substantiels rapportés chez EMI par les Beatles, Sir Hounsfield n'aurait pas pu développer son appareil TDM...

La TDM est progressivement devenue un outil de diagnostic médical indispensable et sophistiqué, exemple à la Fig. 1. L'examen se fait en quelques secondes et procure une résolution spatiale de l'ordre du millimètre. Le contraste est encodé sur l'échelle conventionnelle d'unités Hounsfield (HU), avec -1000 HU correspondant à de l'air, 0 HU, de l'eau, et jusqu'à 3000 HU pour les os et implants métalliques.



FIGURE 1 – Salle de TDM moderne.

Source : http://www.ma.uni-heidelberg.de/inst/ikr/klinik_CT.html

Atténuation des rayons X

Définissons d'abord la géométrie du système d'imagerie. Soit un appareil capable de balayer l'objet avec un faisceau de rayonnement linéaire infiniment mince. Pour une même projection, l'angle d'incidence est le même pour toutes les lectures de détecteur, voir Fig. 2 (a). Pour une projection, la position de la source et du détecteur varie linéairement pour couvrir l'ensemble du champ de vue. Ayant acquis une projection, on tourne le statif par un petit angle et on refait une acquisition. La tranche à reconstruire se situe dans le plan xy , est carrée et centrée à l'origine. La source et le détecteur sont dans ce même plan. Le paramètre $t \in [0, 1]$ désigne la position relative du détecteur lors d'une lecture d'atténuation. Soit $\phi \in [0, 2\pi[$ l'angle du statif lors de l'acquisition. Ces deux variables définissent l'espace du sinogramme, soit

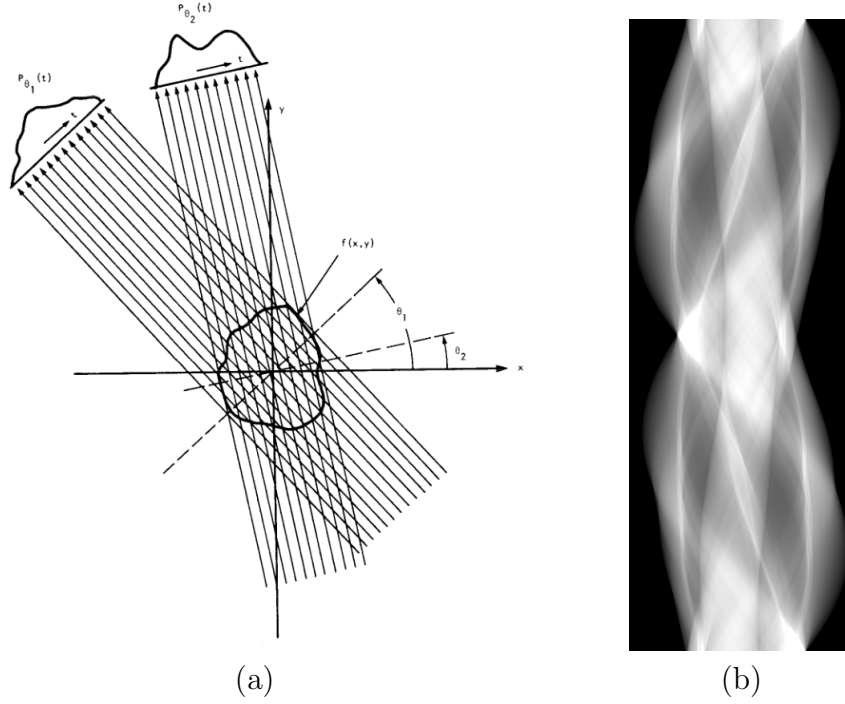


FIGURE 2 – La géométrie d’acquisition à faisceaux parallèles en (a) et un exemple de sinogramme résultant de 256 pixels par 720 projections en (b) (adapté de Kak et. al [1].)

une acquisition complète, voir Fig. 2 (b). Chaque ligne du sinogramme correspond à une projection à un angle donné.

Un appareil clinique utilise un faisceau en éventail et un détecteur en arc de cercle, ce qui implique une géométrie cylindrique un peu plus complexe. Ici, nous simplifions le problème avec une géométrie à rayons parallèles (Fig. 2) qui correspond en fait assez bien à l’appareil développé par Hounsfield.

La reconstruction tomographique est basée sur un modèle de l’atténuation du rayonnement. La loi de Beer-Lambert est souvent utilisée pour caractériser l’intensité du rayonnement à la sortie de l’objet. Soit une intensité détectée I , qui résulte de l’atténuation de l’intensité incidente I_0 selon :

$$I = I_0 \exp \left[- \int \mu(x) dx \right], \quad (1)$$

où μ est le coefficient d’atténuation linéaire des rayons X pour cette énergie du faisceau et x est la distance traversée le long du parcours du rayon. Comme l’image recherchée est discrétisée, on écrira plutôt :

$$I = I_0 \exp \left[- \sum \mu(x) \Delta x \right], \quad (2)$$

Ce problème peut être exprimé sous forme linéaire :

$$\ln \left(\frac{I_0}{I} \right) = \sum \mu(x) \Delta x, \quad (3)$$

Les données de projection fournies par l'appareil sont représentées par la partie gauche de l'équation et $\mu(x)$ est la fonction recherchée.

Questions

1. Supposons, pour simplifier, que le corps humain est fait d'eau pure. Grâce à la base de données XCOM du NIST¹, trouvez le μ de l'eau pour des photons de 60 keV, à une masse volumique de 1.0 g/cm³. Quelle fraction du signal subsiste après avoir traversé 5 cm d'eau ; 20 cm d'eau ?

Reconstruction par rétroprojection

Une façon simple de reconstruire les images de projection vers des images TDM est d'étaler les valeurs des pixels des projections sur la grille des voxels de l'image, comme on étale du beurre d'arachides sur une rôti. Il faut simplement respecter les trajectoires des rayons qui passent à travers l'objet, paramétrées par t et ϕ . Par convention, les entrées du sinogramme sont appelées pixels, car le détecteur est soit en 1D, soit en 2D. Les éléments de l'image reconstruite sont appelés voxels car la reconstruction de l'objet 3D est possible en considérant plusieurs tranches (3D).

Deux fichiers de projections vous sont fournis, en format texte. Chaque ligne de ces fichiers contient toutes les lectures de l'atténuation totale $\ln(I_0/I)$ pour un angle d'incidence, dans la géométrie à faisceaux parallèles de la figure 2. Le premier fichier est `sinogram-password.txt` et correspond au sinogramme d'un mot de passe imprimé blanc sur fond noir. Le second fichier est `sinogram-patient.txt` et contient des projections de type médical. En faisant une reconstruction correcte du mot de passe, vous pouvez accéder à l'URL : <https://physmed.chudequebec.ca/XXXXX/> qui contient le fantôme numérique qui vous permettra de comparer vos reconstructions avec les images d'origine (remplacer XXXXX par le mot de passe de l'image reconstruite). Un autre fichier (`angles.txt`) contient la liste des angles du détecteur, qui est la même pour les deux sinogrammes.

Un squelette du programme est fourni : `geometry.py` contient les paramètres stockés et quelques transformations géométriques de base pour le projet. `util.py` contient des routines utilitaires. `reconstruction.py` contient le squelette du programme de rétroprojection le plus simple.

1. <http://physics.nist.gov/PhysRefData/Xcom/html/xcom1.html>

Il existe deux façons de rétroprojeter. L'une d'elles consiste à suivre la trajectoire de chaque rayon (d'un pixel du détecteur jusqu'à la source de rayonnement) et d'ajouter la valeur de l'atténuation totale à chaque voxel traversé de la grille de l'image, initialisée à 0 avant la reconstruction. Cette approche, pixel-driven dans la langue de Shakespeare, exige des calculs relativement complexes. Une approche équivalente, voxel-driven est de considérer un voxel à la fois, et de sommer chaque petite contribution que lui apporte le pixel correspondant de chaque projection. Cette dernière approche est fortement suggérée pour votre code.

Il est aussi suggéré d'écrire une ligne de code qui imprime l'état de la progression de la reconstruction, qui peut prendre quelques minutes.

2. En utilisant le code fourni (ou pas), créer une rétroprojection simple, aussi appelée laminogramme, des données de projection fournies. Notez qu'une approche par slicing pourrait accélérer votre algorithme. Votre code utilisera la méthode du plus proche voisin pour déterminer la valeur de projection à utiliser (le rayon passant par la source et le voxel d'intérêt n'aboutit pas nécessairement au centre des pixels du détecteur). Une fois que vous aurez accès au fantôme numérique, comparez-le à la reconstruction. Qualitativement, quelles différences observez-vous entre l'image reconstruite et l'image de référence (fantôme) ?

Rétroprojection filtrée

Comme vous avez probablement pu le constater, le laminogramme n'est pas la solution exacte au problème de la reconstruction TDM. Il existe toutefois une méthode analytique pour corriger le problème. Le fait d'étaler les projections sur la grille de l'image ajoute beaucoup de composantes de basse fréquence spatiale dans l'image, de sorte qu'une fonction rectangle dans l'objet imagé (dans les projections) devient plutôt de forme s'apparentant à une gaussienne dans le laminogramme.

Il est possible de corriger ce défaut en filtrant les projections une à une avec un filtre passe-haut, avant de faire la rétroprojection. Soit u la fréquence spatiale dans l'espace de Fourier. Un filtre passe-haut rudimentaire s'exprime comme $f(u) = |u|$, de sorte à ce que les hautes fréquences soient mises en valeur. Ce filtre a tendance à amplifier le bruit à haute fréquence mais fera le travail ici ; des filtres de type Butterworth peuvent aussi être utilisés.

Soit $p(t)$ une projection et \mathcal{F} l'opérateur de la transformée de Fourier. L'algorithme pour obtenir des projections filtrées $p_f(t)$ est donc le suivant :

$$P(u) = \mathcal{F}[p(t)] \quad (4)$$

$$P_f(u) = P(u) \times |u| \quad (5)$$

$$p_f(t) = \mathcal{F}^{-1}[P_f(u)] \quad (6)$$

Notez que l'application du filtre pourrait aussi se faire dans le domaine spatial par une opération de convolution, plus lourde numériquement que l'opération de multiplication équivalente dans le domaine fréquentiel. Par contre, rester dans le domaine spatial serait contraire à l'esprit de ce TP, qui est d'explorer la transformée de Fourier.

3. En utilisant la librairie `numpy.fft`, implémentez le filtre passe-haut proposé pour filtrer le sinogramme une projection à la fois². Affichez et sauvegardez l'image du sinogramme filtré. Qualitativement, quelles sont les principales différences entre le sinogramme fourni et filtré?
4. Ayant en main le sinogramme filtré, effectuez une rétroprojection filtrée, en récupérant idéalement le code précédent du laminogramme. Comparez l'image obtenue avec le fantôme. Quelles différences observez-vous? Où sont situées les erreurs et que diriez-vous de la fréquence spatiale des erreurs?
5. Tentons d'améliorer la qualité d'image davantage. Ecrivez une nouvelle fonction dans `geometry.py` qui permet d'ajouter une valeur d'atténuation plus précise à chaque voxel lors de la rétroprojection. Au lieu d'identifier le pixel plus proche voisin sur le détecteur de radiation, utilisez l'interpolation linéaire entre deux pixels adjacents sur le détecteur. Comment l'image s'est-elle améliorée? Pour référence future, enregistrez le temps de reconstruction total.

Reconstruction via le théorème de la tranche de Fourier

Il existe une méthode analytique très différente pour obtenir les images en TDM. Elle est basée sur le théorème de la tranche de Fourier (Fourier slice theorem ou Central slice theorem). On ne s'attardera pas à la preuve formelle, mais on va plutôt retenir un résultat utile : lorsqu'une projection est acquise en géométrie à faisceaux parallèles (comme c'est le cas ici), la transformée de Fourier de la projection 1D correspond à une ligne de la transformée de Fourier 2D de l'image recherchée, au même angle d'incidence que celui de la projection, voir Fig. 3 (a).

$$\text{TF} [p(t)_{\phi=\text{const}}] = \text{TF}_{2\text{D}} [\mu(t, \phi = \text{const})] \quad (7)$$

On peut donc peupler l'espace Fourier de l'image $M(u, v)$ avec les lignes de projection $P(t)_{\phi}$ et faire la TF inverse par la suite pour obtenir l'image recherchée. Ceci pose un problème technique non négligeable car l'image recherchée est exprimée sur une grille cartésienne alors que les projections remplissent un espace polaire, avec un échantillonnage tangentiel de plus en plus espacé lorsqu'on s'éloigne de l'origine, voir Fig 3 (b).

6. Ecrivez une nouvelle fonction de reconstruction `reconFourierSlice()` qui effectue la reconstruction via le théorème de la tranche de Fourier. Il faut effectuer la TF 1D de chaque projection du sinogramme et stocker le nouveau sinogramme transformé. Par

2. Indices : lisez bien à quoi sert la fonction `fftshift` et n'oubliez pas que $p(t)$ est une fonction réelle

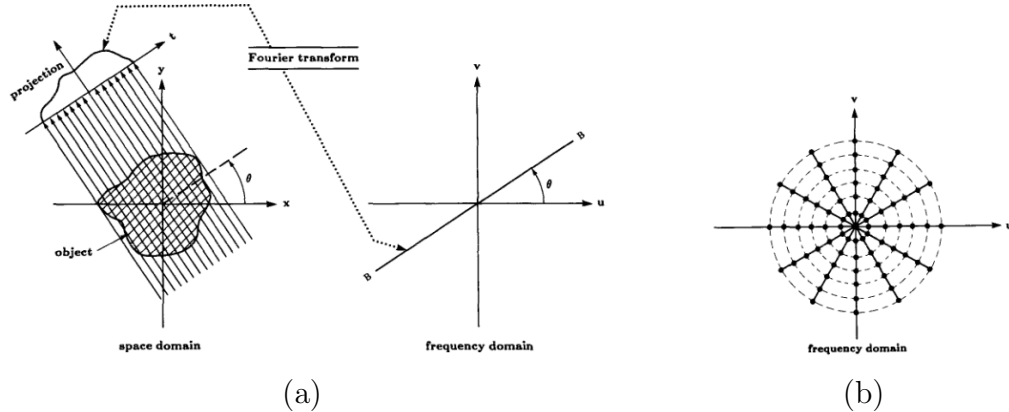


FIGURE 3 – (a) Visualisation du théorème de la tranche de Fourier, (b) échantillonnage tangentiel irrégulier de l'espace de Fourier 2D. Figures adaptées de Kak et. al [1].

échantillonnage du sinogramme, remplissez l'espace de Fourier 2D de l'image à reconstruire. Vous pouvez choisir l'angle approprié par la méthode du plus proche voisin et la position sur le détecteur par interpolation linéaire. Trouvez l'image par TF_{2D}^{-1} . Indice : considérez toutes les données comme des nombres complexes avec l'initialisation suivante `mymatrix = numpy.zeros((a,b), 'complex')` afin d'éviter les changements de taille, puis prenez la partie réelle de l'image finale.

7. Incluez dans le rapport et décrivez brièvement le résultat obtenu. Il est fort probable que l'image soit loin de vos attentes, même si le code est correct. Notez également le temps de reconstruction.
8. Tentez de reconstruire l'image en soumettant une grille d'image deux fois plus dense dans chaque direction (doubler le nombre de voxels et réduire leur taille deux fois dans le fichier `geometry.py`). Quelles améliorations observez-vous ? L'image est-elle acceptable pour un diagnostic médical ? Basez votre discussion sur la définition-même de la TF et sur le théorème de Nyquist-Shannon.
9. Comparez les temps de reconstruction de la rétroprojection filtrée, la reconstruction TF_{2D} et cette dernière avec sur-échantillonnage. Discutez de la taille des boucles `for` imbriquées dans ces trois cas et/ou de l'utilité du slicing si applicable.
10. **BONUS :** Implémentez une fonction d'interpolation bi-linéaire pour la méthode TF_{2D} , qui se fie à quatre voisins du voxel échantillonné. Observez-vous une amélioration ? Qu'en est-il du temps de reconstruction ?

Instructions pour la remise

Le travail devra être complété en équipe sous format de cahier de bord jupyter (`.ipnb`) et remis dans la boîte de dépôt ENA créée à cette fin. Ce document contiendra toutes informations pertinentes permettant au lecteur d'apprécier vos résultats et conclusions, incluant le code Python utilisé et d'éventuelles références bibliographiques. La qualité de la présentation est très importante (utilisation de sections, de graphiques appropriés, de mise en contexte,

etc.). Ajouter également une version pdf de votre notebook.

Prenez soin de bien indiquer votre (ou vos) nom(s) dans le cahier de bord. Pour faciliter la tâche de classification, utilisez la nomenclature suivante pour le fichier transmis (un seul) : TPn_nom1.ipnb ou TPn_nom1_nom2.ipnb

Références

- [1] M. Slaney and A. Kak, *Principles of computerized tomographic imaging*. IEEE Press, 1988. [Online]. Available : <http://www.slaney.org/pct/index.html>