

GBM8378 – Laboratoire 1 : Rayon-X

Auteur original du laboratoire : Simon Dubeau. Version: Hiver 2018

Introduction	1
Théorie	2
Reconstruction par rétroprojection	3
Théorème de la tranche centrale	4
Bruit	5
Commandes Matlab	7
Travail à effectuer	9
Questions préliminaires – 5pts	9
Projection et rétroprojection – 2pts	9
Rétroprojection filtrée – 5pts	10
Performance de la rétroprojection filtrée – 2pts	10
Théorème de la tranche centrale – 3pts	10
Analyse du bruit – 3pts	10
Critères d'évaluation	11

Introduction

L'expression *tomographie axiale* décrit en général un ensemble de techniques numériques servant à reconstruire l'image de coupe d'un objet à partir d'un ensemble de projections unidimensionnelles. L'assemblage de plusieurs coupes permet en outre de reconstruire l'objet en 3D. Cette technique est utilisée notamment par les appareils de mesure (*scanners*) de tomographie par rayon X (*CT scan*) et en tomographie par émission de positons (*PET scan*). En effet, le *CT* (et le *PET scan*) fait l'acquisition de rayons X (positons pour le *PET*) traversant le corps dans plusieurs directions de l'espace : ce sont les projections. La figure 1 présente une image reconstruite de *CT* et de *PET*, ainsi que la combinaison des deux.

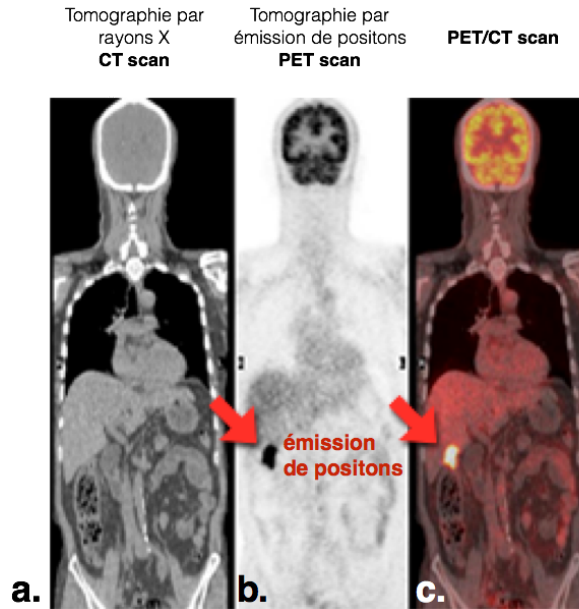


Figure 1: a. Tomographie par rayon X : Les os et les organes apparaissent clairement. b. PET scan. Les cellules présentant une activité élevées (e.g. cellules cancéreuses) présentent une forte émission de positons. c. La combinaison des deux permet de révéler l'organe affecté.

Dans ce laboratoire, vous allez réaliser les algorithmes pour reconstruire les images 2D à partir des projections mesurées par un scanner à rayon X. Les projections tomographiques peuvent être obtenues en utilisant une géométrie à rayons parallèles (scanner de première génération) ou à rayons divergents. Dans le cadre de ce laboratoire, pour simplifier, nous allons étudier principalement les principes de tomographie axiale en utilisant les rayons parallèles.

Durant ce laboratoire vous apprendrez à :

- Reconstruire des images par tomographie axiale
- Comprendre l'effet du filtrage sur la reconstruction des images
- Observer l'effet du bruit, du temps d'acquisition des images et du nombre d'angles différents utilisés sur la qualité des images reconstruites

Théorie

En *tomographie axiale*, l'objet à imager est modélisé par une fonction bidimensionnelle $f(x,y)$ représentant la propriété physique de l'objet qui sera mesurée. $f(x,y)$ peut être la radioactivité locale (médecine nucléaire), le temps de transit d'une onde acoustique (tomographie axiale par ultrasons, une méthode rarement utilisée toutefois), la densité de protons (imagerie par résonance magnétique nucléaire) ou le coefficient d'atténuation aux rayons X.

En termes mathématiques, la projection tomographique illustrée en figure 2 est le résultat de l'intégrale de ligne (le long de la direction de projection) de $f(x,y)$. Le signal mesuré est lié à l'intensité du faisceau de rayons X ayant traversé les tissus, soit :

$$I_d = I_0 \exp \left[- \int_0^d \mu(s; \bar{E}) ds \right] \quad (1)$$

$$\ln \left(\frac{I_d}{I_0} \right) = - \int_0^d \mu(s; \bar{E}) ds \quad (2)$$

où : I_0 , I_d sont respectivement l'intensité du rayon incident et émergent, \bar{E} est l'énergie effective du rayon et $\mu(s)$ est le coefficient d'atténuation linéaire (en m^{-1}).

Pour effectuer la reconstruction tomographique, il est nécessaire d'avoir des projections le long d'un axe l pour différents angles θ . La matrice $f(l, \theta)$ représentant les projections pour différents angles est appelée **sinogramme**.

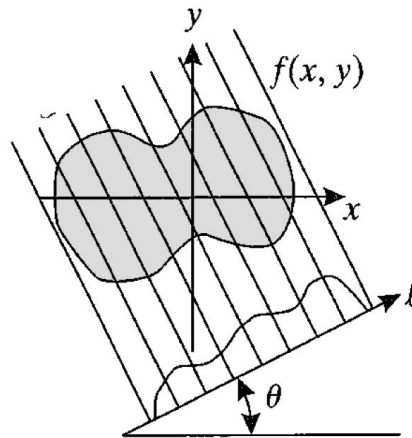


Figure 2 - Exemple d'une projection

Reconstruction par rétroprojection

La rétroprojection consiste à reprojeter la projection obtenue pour un angle θ sur toute la surface imagée. La reconstruction d'image par rétroprojection consiste à reconstruire une image en additionnant les images obtenues par rétroprojection pour différents angles.

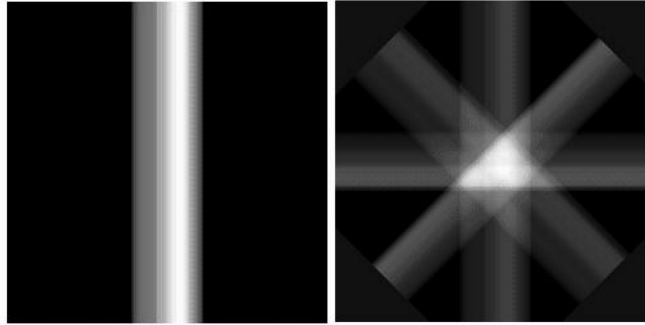


Figure 3 - Exemple d'une rétroprojection avec 1 et 4 angles

Théorème de la tranche centrale

Ce théorème est connu sous le nom de *Fourier Slice Theorem* dans les publications anglophones. C'est sur ce théorème que reposent les modèles analytiques de reconstruction tomographique.

Ce théorème fait un lien entre le domaine spatial et le domaine spectral (après transformée de Fourier). La figure 4 illustre la transformée de Fourier 2D de quelques images.

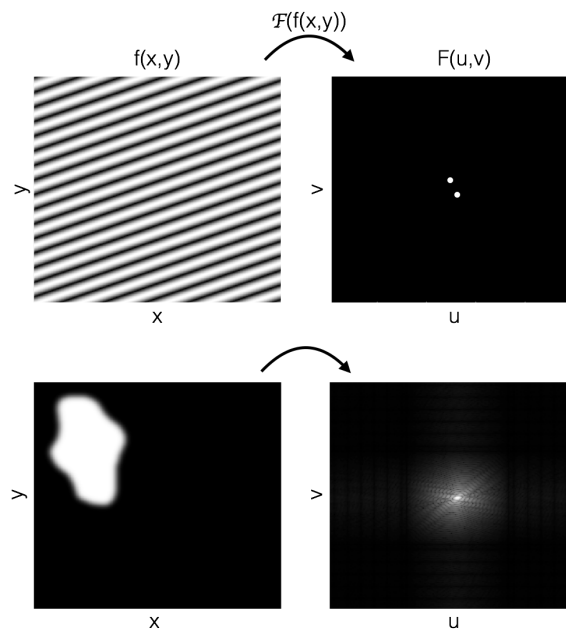


Figure 4 - Image et sa transformée de Fourier en utilisant la fonction Matlab FFT2

Le théorème de la transformée de Fourier d'une projection nous dit que la transformée de Fourier (1D) d'une projection de $f(x,y)$ sur un axe d'orientation θ est identique à la tranche centrale d'angle θ de la transformée (2D) de $f(x,y)$. Par *tranche centrale*, on entend ici l'ensemble des valeurs que prend la transformée de Fourier sur une droite qui passe par l'origine du domaine spectral.

Le théorème est illustré en figure 5 avec : ℓ La projection d'angle θ , Q sa transformée de Fourier 1D, $f(x,y)$ le domaine spatial et $F(u,v)$ le domaine spectral (transformée 2D). La figure 6 illustre la position de 8 projections (Q) dans le domaine de Fourier (u,v). On peut constater sur cette représentation que le centre de l'espace de Fourier (basses fréquences) est suréchantillonné par rapport au bords. Afin de réduire l'effet des basses fréquences, chaque projection est filtrée par une fonction rampe $|\omega|$ dans le domaine spectral (voir le chapitre 6.3.3, Prince & Links) (rétroprojection filtrée). Dans le cas de ce laboratoire, le filtrage fréquentiel est effectué à l'aide d'une convolution dans le domaine spatial.

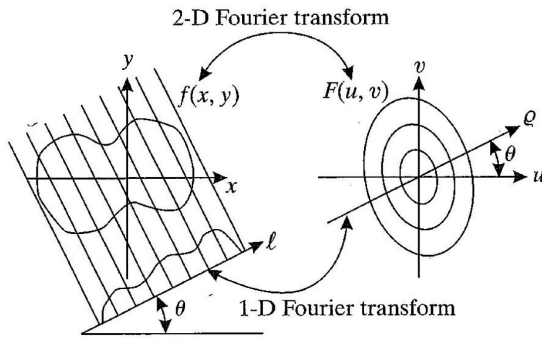


Figure 5 - Représentation du théorème de la tranche centrale

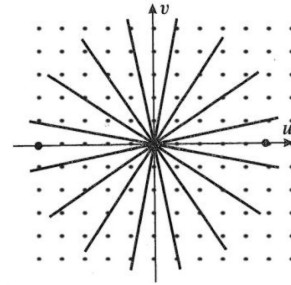


Figure 6 - Domaine spectral de la transformée discrète et des projections

Bruit

En tomographie, la probabilité qu'un photon soit détecté par un détecteur peut être modélisée par un processus de Poisson de moyenne $\mu_g = \lambda$ et de variance $\sigma_g^2 = \lambda$. Le rapport signal sur bruit (RSB) est donc donné par :

$$RSB = \frac{\mu_g}{\sigma_g} = \frac{\lambda}{\sqrt{\lambda}} = \sqrt{\lambda} \quad (3)$$

On peut aussi définir le RSB différentiel :

$$RSB_{diff} = \frac{CI_{ap}}{\sigma_b(A)} \quad (4)$$

où C est le contraste, I_{ap} est l'intensité de l'arrière-plan, et $\sigma_b(A)$ est l'écart-type de l'arrière-plan mesuré sur une aire de même dimension que la cible. Le contraste d'une image étant défini comme :

$$C = \frac{I_{cible} - I_{ap}}{I_{ap}} \quad (5)$$

Modélisation du bruit

Nous devons modéliser un signal d'intensité et de variance $\mu_g = \sigma_g^2 = I_{in}$ provenant d'une distribution de Poisson. La fonction matlab `random('poiss',...)` peut simuler une image

provenant d'un processus de Poisson. En faisant varier λ , la ligne suivante permet de modéliser une image obtenue suivant des temps d'exposition plus ou moins longs :

```
Imagebruitee=random('poiss',image*lambda,size(image));
```

Sensibilité/spécificité

Pour déterminer la sensibilité et la spécificité d'un appareil, il est préférable de d'abord établir le tableau de contingence (Tableau 1). Pour ce faire, il faut fixer un seuil puis évaluer le nombre de vrais et faux positifs ainsi que le nombre de vrais et faux négatifs.

Tableau 1: Tableau de contingence

		Réalité	
		Malade	Sain
Diagnostic	Malade	Vrai positif (A)	Faux positif (B)
	Sain	Faux négatif (C)	Vrai négatif (D)

Il est ensuite possible de calculer la proportion de patient malade que le test diagnostique identifie comme étant malade (sensibilité) et la proportion de patient sain que le test diagnostique identifie comme étant sain (spécificité).

$$\text{Sensibilité} = \frac{A}{A+C} \quad \text{Spécificité} = \frac{D}{B+D}$$

La précision du diagnostic peut aussi être calculée. La précision du diagnostic correspond à la proportion de diagnostics valides.

$$\text{Précision} = \frac{A+D}{A+B+C+D}$$

Enfin, en faisant varier le seuil, il est possible d'effectuer une analyse ROC afin d'optimiser le seuil. Pour tracer une courbe ROC (figure 6), il suffit de calculer la sensibilité et la spécificité à différents seuils puis de les rapporter l'un en fonction de l'autre.

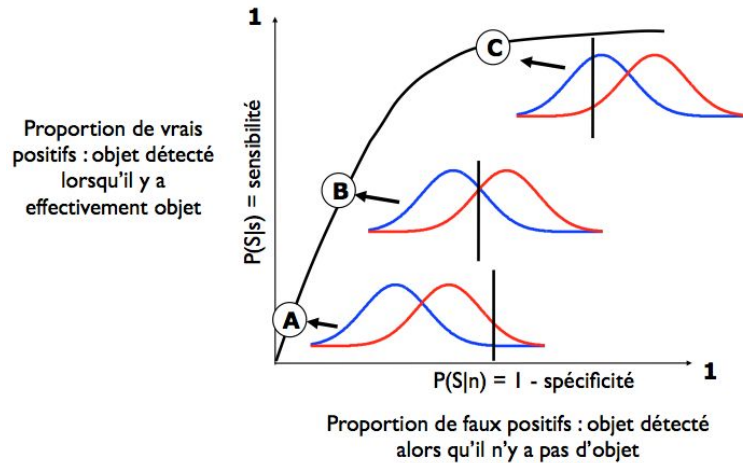


Figure 7: Courbe ROC

Commandes Matlab

Chargement des fonctions

Plusieurs fonctions utiles pour le laboratoire se trouvent dans le fichier `labCTfonction.m`. Pour mettre ces fonctions en mémoire, exécuter la ligne de commande suivante :

```
[modele_bruit dosinogramme retroprojection filtrage Contraste] = labCTfonction;
```

Ensuite, appeler les fonctions de la manière habituelle sous Matlab.

FFT

Les fonctions `fft` (Fast Fourier Transform) et `ifft` de Matlab permettent respectivement d'effectuer la transformée de Fourier discrète d'un signal ainsi que la transformée inverse.

$$\text{fft} : X(k) = \sum_{j=1}^N x(j) w_n^{(j-1)(k-1)} \quad (6)$$

$$\text{ifft} : x(j) = \frac{1}{N} \sum_{k=1}^N X(k) w_n^{-(j-1)(k-1)}, \text{ où } w_n = e^{-2\pi i/N} \quad (7)$$

Ces fonctions utilisent un algorithme qui fonctionne pour des vecteurs de taille 2^n . Si les vecteurs n'ont pas cette taille, la fonction ajoutera des zéros à la fin du vecteur (*zero padding*) pour obtenir un vecteur de la taille appropriée. La FFT permet de travailler avec un signal temporel échantillonné uniformément dans le temps x_j pour en obtenir une représentation spectrale discrète X_k . Pour un signal temporel discret x_j de N échantillons à une fréquence d'échantillonnage F (pas de $1/F$), la FFT produit la représentation spectrale de N échantillons échantillonnés à un pas de F . La FFT produit donc un signal allant de F_0 à F_{N-1} .

La propriété de symétrie de la FFT implique que pour un signal réel X_n , $X_n^* = X_{-n}$ où X_n^* est la conjuguée complexe de X_n . De plus, la périodicité de la FFT donne que $X_k = X_{k+N}$. Ainsi, le

signal échantillonné de $X_{N/2}$ à X_{N-1} est l'équivalent du signal de $X_{-N/2}$ à X_{-1} . La fonction `fftshift` permet de prendre en compte ces propriétés de la FFT, et déplace les valeurs d'un vecteur pour avoir un vecteur qui va de $X_{-N/2}$ à $X_{N/2-1}$ au lieu de X_0 à X_{N-1} . La fonction `ifftshift` fait l'opération inverse. On constate que la plus basse fréquence spectrale est $F_{N/2}$. Ainsi, un signal x_k réel de N échantillons échantillonné à une fréquence de F Hz a comme équivalent dans le domaine spectral une série X_N allant de $-F_{N/2}$ Hz à $F_{N/2}$ Hz. Pour une image en deux dimensions, la fonction `fft2` effectue la transformée de Fourier discrète 2D d'une image.

conv2

La fonction `conv2(A,B)` permet d'effectuer la convolution de deux matrices. Si B est un vecteur, la fonction effectue la convolution dans une seule dimension.

Affichage d'une image

La fonction `imagesc` affiche une image dans Matlab. Par défaut, le point $f(0, 0)$ se trouve en haut à gauche de la figure. La commande `axis xy` permet de placer le point $f(0, 0)$ en bas à gauche de la figure. La commande `colorbar` affiche la barre chromatique à droite de la figure. La commande `colormap gray` affiche la figure en noir et blanc ce qui permet d'obtenir des résultats satisfaisants lors de l'impression en noir et blanc.

Affichage de plusieurs figures

La fonction `subplot` permet d'afficher plusieurs graphiques à l'intérieur d'une même figure.

```
figure % Créer la figure
subplot(2,3,1) % Accède au graphique 1 (en haut à gauche) sur une figure divisée
en 2 rangées et 3 colonnes
plot(rand(2,2)), % Créer la figure
subplot(2,3,2) % Accède au graphique 2 (en haut au centre) sur la % figure divisée
en 2 rangées et 3 colonnes
plot(rand(2,2)), % Tracer la courbe
```

Projection avec un faisceau en éventail (fanbeam)

Les fonctions `fanbeam` et `ifanbeam` permettent de modéliser l'acquisition et la reconstruction d'images. Le code suivant permet de simuler la reconstruction d'une image à l'aide d'un fan beam CT scan.

```
angle = 5; %'FanRotationIncrement'
spacing = 2; %'FanSensorSpacing'
filt = 'Hann';
fb = fanbeam(im2,150,'FanRotationIncrement',angle,'FanSensorSpacing',spacing);
% Note fb: fb(i,j): j: les angles de rotation, i: les détecteurs
ifb = ifanbeam(fb,150,'FanRotationIncrement',angle,'FanSensorSpacing',...
spacing,'filter',filt);
```


Travail à effectuer

Complétez le Notebook Jupyter.

N.B: Pour pouvoir exécuter Matlab dans un notebook Jupyter, il faut installer Anaconda, installer le kernel de Matlab. Les postes du Lab à Poly sont déjà configurés pour cela. Toutefois, si vous souhaitez travailler sur les labs de votre ordinateur personnel, voici les indications à suivre: Pour ce faire, je vous invite à suivre la slide #5 sur ce lien (seulement la section "MATLAB or Octave Kernel"): <https://slides.com/agahkarakuzu/qmrlabjn1#/0/4>

Critères d'évaluation

- Le rapport (en format *.pdf*) doit être déposé sur *Moodle*.
- Le code (en format *.zip* ou *.ipynb*) doit être déposé sur *Moodle*.
- Pénalité de 5 % par jour de retard.
- Présentation et langue.