Reconstruction d'images tomographiques

20 juin 2024

1 Introduction

Un tomographe correspond à une source de rayons X qui traversent un corps humain ou un objet. On ne connait pas la composition interne de ce corps ou de cet objet, mais l'atténuation des rayons va nous fournir des indices sur cette composition. En effet les tissus traversés atténuent les rayons différemment suivant leur densité.

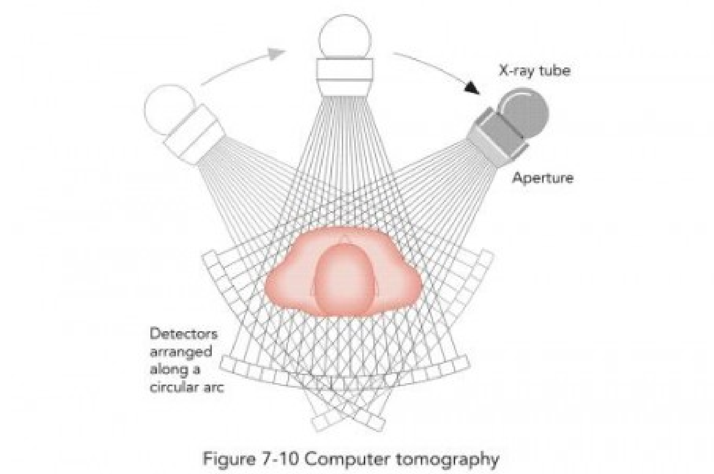


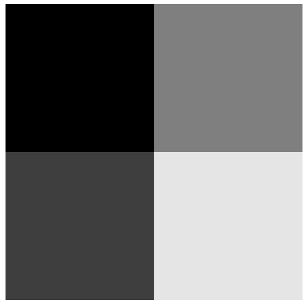
Figure 1 – Source : Les petites curies du net

Au moment de l'acquisition, il est impossible de connaître la densité des tissus de manière directe : le tomographe ne mesure que la projection de ces informations, par cumul des densités rencontrées le long de rayons. Dans notre modèle, ces rayons sont parallèles à une direction fixée (ce qui est légèrement différent de l'illustration de la figure 1, où les rayons sont issus d'une source ponctuelle).
Lorsque l'acquisition dans une direction est achevée, le tomographe se déplace afin de passer à une direction différente des rayons. La densité doit être retrouvée à partir des valeurs projetées et c'est ce que nous nous proposons de faire au cours de ce stage.

Les volumes étudiés en tomographie médicale sont 3D au sens où le tomographe permet de retrouver la densité des tissus en tout point d'un volume 3D, mais dans le cadre de notre travail, nous allons nous familiariser avec le principe de la tomographie en nous restreignant au cas 2D, en reconstruisant des images planes. Dans ce cas nous retrouvons l'information d'une zone 2D en la traversant avec des rayons du plan 2D.

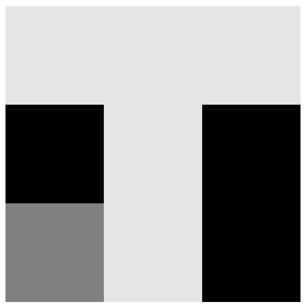
2 Simulation de notre tomographe 2D

Considérons une première image 2D à tomographier (voir figure 2). Cette image est discrète au sens où elle correspond à un tableau 2D de valeurs. Commençons par nous familiariser avec la notion de projection des intensités de cette image dans une direction donnée.



0	50
25	90

FIGURE 2 – Notre première image 2D à tomographier et sa vue comme un tableau de valeurs de densités. Cette image est composée de 2 lignes et de 2 colonnes et la densité maximale est ici de 100.



90	90	90
0	90	0
50	90	0

FIGURE 3 – Une autre image un peu plus grosse (3 lignes et 3 colonnes).

2.1 Projections dans la direction horizontale

Pour chaque ligne de votre image, vous allez cumuler les valeurs de densités rencontrées sur les pixels. Mathématiquement, on dit que l'on réalise une intégration.

— Quelles valeurs cumulées obtenez-vous et comment proposez-vous de les stocker informatiquement ?

2.2 Projections dans la direction verticale

Pour chaque colonne de votre image, vous allez également faire le cumul des valeurs rencontrées et stocker les résultats obtenus.

2.3 Sensibilisation au problème à résoudre

Le principe de la tomographie et de la transformée de Radon consiste à retrouver l'image 2D ou une approximation de cette image à partir des projections que vous avez calculées.

— Pensez-vous avoir assez de projections en utilisant uniquement les projections horizontales et verticales pour reconstruire la première image (Figure 2) ?
— Est-ce également le cas avec la seconde image tomographique (Figure 3) ?
— Avez-vous une idée du nombre de directions de projections qu'il faudrait en pratique pour reconstruire une image de grande taille ? Rassurez-vous, il est possible de produire un résultat approché avec un nombre moins important de projections.

2.4 Projections dans une autre direction

Comme nous travaillons avec des images discrètes, nous vous proposons de travailler également avec des rayons discrets le long desquels nous allons réaliser nos projections de densités, par cumul des valeurs rencontrées.

2.4.1 Projections dans la direction diagonale

Par exemple, nous allons modéliser les rayons de pente 1 en nous déplacant en diagonale dans le tableau de valeurs, en partant d'un pixel de la première ligne ou de la première colonne.

— Dessinez toutes les diagonales de pente 1 que vous voyez dans l'image.
— Quelles sont les valeurs cumulées obtenues dans la direction diagonale de pente 1 ?
— Même question pour la seconde direction diagonale (pente -1).
— Combien de projections diagonales avez-vous en tout et comment proposez-vous de les stocker informatiquement ?

2.4.2 Projections dans la direction de pente n/m

Une pente n/m signifie que l'on se déplace de n colonnes pour m lignes. Pour parcourir un rayon de pente n/m avec n et m deux nombres entiers et $n < m$, nous proposons de partir d'un pixel dans la première colonne de l'image ou dans la première ligne. De manière préalable, nous réalisons une division entière de n par m de manière à obtenir q tel que $m = n * q + r$ avec $r < n$. Etant donné un pixel de départ, on va visiter un certain nombre de pixels par ligne de l'image. Sur chaque ligne rencontrée, on se déplace, en les visitant, de q pixels sur la droite, avant de se décaller sur la ligne suivante. Si le reste r n'est pas nul, alors toutes les $n + 1$ lignes, le nombre de pixels visités ne sera que de r (au lieu des q pixels rencontrés sur chaque ligne).

— Avez-vous compris comment vous promener sur un rayon de pente $3/5$?
— Pour une pente donnée, combien de projections avez-vous en tout et comment proposez-vous de les stocker informatiquement ?
— Quelles valeurs obtenez vous pour les projections de pente $3/5$?
— Comment faire pour se déplacer sur une droite de pente $n > m$ avec $n > m$?
— Comment faire pour se déplacer sur une droite de pente négative ?
— Si vous possédez des rudiments de programmation, réalisez le programme permettant de construire les projections d'une image dans une direction de pente donnée.

3 Reconstruction de l'image à partir des projections

Nous allons maintenant imaginer comment retrouver la valeur des pixels de notre image à partir des projections dans un ensemble de directions. Pour cela nous allons d'abord nous restreindre à un problème plus simple, qui sera ensuite utile pour imaginer la résolution finale.

3.1 Résolution d'un problème simplifié

Dans un premier temps nous allons d'abord déterminer pour chaque pixel une valeur maximum de sa valeur.

— Etant donnée la valeur de densité cumulée le long d'un rayon, quelle valeur maximale proposez-vous pour chacun des pixels rencontrés par le rayon ?
— Etant donné un pixel couvert par plusieurs rayons, quelle valeur maximale proposez-vous pour le pixel ?
— Pensez-vous qu'un second passage puisse être utile pour affiner cette valeur maximale ?
— Faites le calcul des maxima pour tous les pixels de l'image 3, à partir des projections dans les directions horizontales, verticales et diagonales que vous avez calculées précédemment.
— Si vous possédez des rudiments de programmation, réaliser le programme permettant de reconstruire la valeur maximale des pixels d'une des images mystères fournies (de taille L lignes et L colonnes), à partir des projections fournies dans un ensemble de directions.
— Visualisez les valeurs max obtenues (nous vous aiderons pour cela à construire une image *pgm*). Il s'agit d'une première approximation de l'image.

3.2 Exploration de plusieurs pistes pour reconstruire l'image

Il nous reste à présent à reconstruire les valeurs des pixels de l'image, en respectant les valeurs maximales calculées précédemment ainsi que les valeurs des projections.

A vous de jouer ! Commencez par réfléchir au problème sur papier et des exemples simples

4 Fichiers fournis et éléments de code en Python

Les images seront fournies au format *pgm*. Il s'agit d'images de niveau de gris. Les fichiers de projections tomographique sur lesquels nous travaillerons ont la forme suivante. Pour chaque rayon :

— Pente du rayon sous forme de 2 entiers (seul le second peut être négatif),

— Coordonnées du pixel de départ

Voici des éléments de code qui vous aideront d'une part à charger une image *.pgm* dans des tableaux et d'autre part à charger les projections associées à une image.

```
import numpy as np #numpy est la librairie qui va gérer les tableaux
from PIL import Image #librairie qui va gérer la lecture et l'écriture d'images

#lire une image pgm binaire sous forme d'un tableau numpy
image = np.array(Image.open('testbin.pgm'))
# ou bien (si l'image est en couleur ou dans un autre format)
image = np.array(Image.open('miniminifrog.jpg').convert('L'))


#lire un fichier de projection
#cela donne un tableau où les deux premières colonnes sont la pente
#les deux colonnes suivantes sont les coordonnées du point de départ
# qui est un point sur le bord de l'image.
#la dernière colonne est la valeur de la somme
projections = np.loadtxt("projections.asc")

#pour écrire les valeurs de l'image reconstruite on crée un tableau
#numpy de la bonne taille n
n=4
image = np.zeros((n,n))


#Exemple pour écrire la valeur 2 à la coordonnée (0,0) on fait:
image[0,0] = 2
#Exemple pour écrire la valeur 255 à la coordonnée (1,2) on fait:
image[1,2] = 255

#exemple pour parcourir les lignes et colonnes de l'image
for i in range(0,n):
    for j in range(0,n):
        image[i,j] = image[i,j] # à remplacer par vos instructions


#sauver le tableau dans une image
im = Image.fromarray(image)
im.save("resultat.pgm")

# pour écrire dans un fichier
# (par exemple pour écrire les projections)
fichier = open("projections.asc", "w")
val=0
fichier.write(f"J'écris la valeur de val {val}\n")
fichier.close()
```