IMED : Questions ouvertes

Comment rehausser une image radiographique pour : (a) visualiser de petites structures du poumon ? (b) rehausser les os ? Connaissez-vous d'autres décompositions multi échelles avec lesquelles vous pourriez jouer ? Connaissez-vous d'autres algorithmes de rehaussement d'image ?

Rehaussement d'images

Egalisation d'histogramme

L'égalisation d'histogramme est une technique consistant en l'étirement des valeurs de l'histogramme d'une image de manière à accroitre le contraste de celle-ci. Cette méthode exploite le fait que l'on a souvent des images ayant des histogrammes avec un grand nombre de valeurs centrées autour de quelques intensités. De ce fait, on va chercher à avoir une répartition plus uniforme de ces valeurs sur la totalité de la plage des intensités de l'image. On fait ainsi correspondre une ancienne distribution à une nouvelle plus large et plus uniforme. Pour se faire, on va calculer l'histogramme cumulé croissant de l'image :

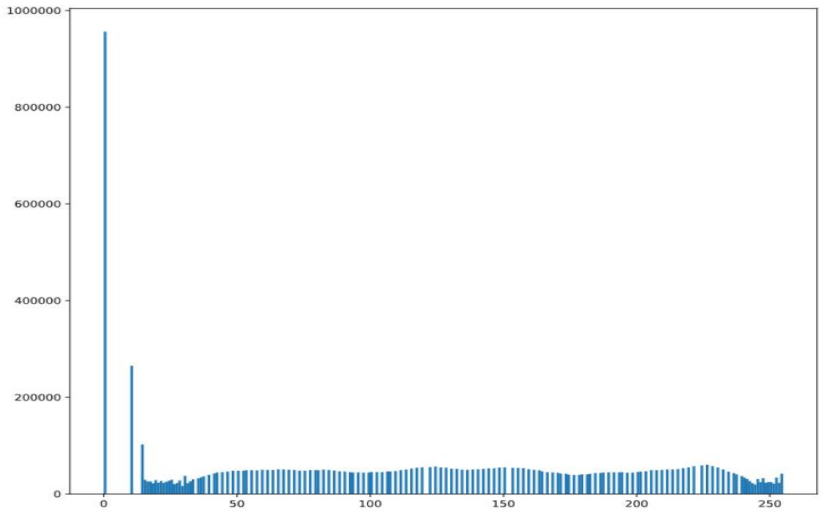
$$H'(i) = \sum_{0 \leq j < i} H(j)$$

Où H(i) est l'histogramme de l'image et H'(i) la distribution cumulée

Ensuite, on normalise H'(i) de manière à avoir la valeur maximum qui coïncide avec la plus haute valeur possible de l'image (255 pour des images de 8 bits). Enfin, on obtient la nouvelle intensité du pixel de la manière suivante :

Nouvelle valeur = H'(ancienne valeur)

Dans la suite, on a décidé de rééchelonner les valeurs de l'image entre 0 et 255 grâce à la fonction 'normalize ' d'openCV de manière à pouvoir appliquer la plus large gamme d'algorithme sur l'image. Cette fonction va faire correspondre la plus faible intensité de l'image à la valeur 0 et la plus forte à la valeur 255 de manière à obtenir une image sur 8 bits.

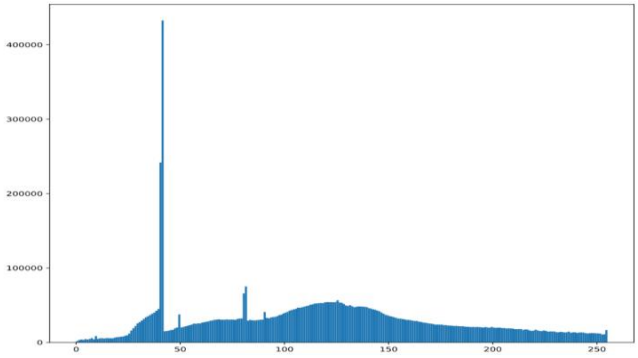


On constate que l'histogramme est plus étiré avec des intensités répartis plus uniformément sur la plage des valeurs possibles. Cependant, l'image est assez bruitée. On peut le constater car l'histogramme n'est pas 'plein', on a des sauts entre les barres du graphe. Visuellement, ce bruit se caractérise par le fait qu'on a du mal à distinguer la colonne vertébrale en dessous des poumons. Il y a un effet d'éblouissement. Lorsqu'on essaye d'estimer l'écart type du bruit gaussien avec la fonction 'estimate_sigma' de skimage, on obtient une valeur de 1.5256 alors qu'on obtenait une valeur de 0.017 sur l'image avant la normalisation d'histogramme. Cela confirme que l'image est plus bruitée.

En réalité, cette image se prête peu à la normalisation d'histogramme car toutes les intensités ne sont pas agglutinées ensemble. A l'inverse, comme le montre l'histogramme de l'image d'origine, on a beaucoup de pixels à faible intensité. C'est lié au fait que les bords de l'image sont constitués de pixels noirs.

CLAHE (adaptive histogram equalization with contrast limiting)

Pour résoudre ce problème on peut appliquer ' l'adaptive histogram equalization with contrast limiting ' (CLAHE). Cet algorithme fonctionne en divisant l'image en sous blocs et en appliquant l'égalisation d'histogramme à chaque bloc. Cela a tendance à accroitre le bruit dans les zones homogènes de l'image. C'est pour résoudre ce problème qu'on a introduit le concept de 'contrast limiting'. Cela consiste à définir un seuil au delà duquel les valeurs d'intensité sont redistribuées uniformément entre les bins de l'histogramme. Cela aura pour conséquence de faire augmenter le nombre de valeur par bin à nouveau au delà du seuil. Cependant on peut appliquer ce processus récursivement jusqu'à ce que le surplus devienne négligeable. Enfin l'algorithme applique une interpolation bilinéaire pour effacer les artefacts au niveau des frontières entre blocs.



L'image semble plus contrastée. On constate que l'on voit mieux les os et notamment les côtes. Les détails de la colonne vertebrale sont plus apparents. Les bronches sont mieux contrastées par rapport au reste du poumon. Globalement l'image semble être plus facile à étudier pour un radiologiste par exemple. Par ailleurs, On note que l'histogramme a moins de valeurs extrêmes et que les valeurs sont plus centrées. L'histogramme est également plus 'plein'. Néanmoins, on a beaucoup de pixels ayant une intensité proche de 40. Ce sont les pixels noirs qui sur l'image de départ étaient localisés au bord de l'image. Cette valeur aberrante dans l'histogramme a tendance à accroitre le bruit de l'image. C'est ce que l'on constate en estiment l'écart type du bruit gaussien qui est ici de 3.08.

Visuellement, on remarque beaucoup de bruit au niveau de la colonne vertébrale sous les poumons, mais également plus haut, au niveau de coup. Nous allons à présent tenter de réduire ce bruit grâce à la décomposition en ondelettes.

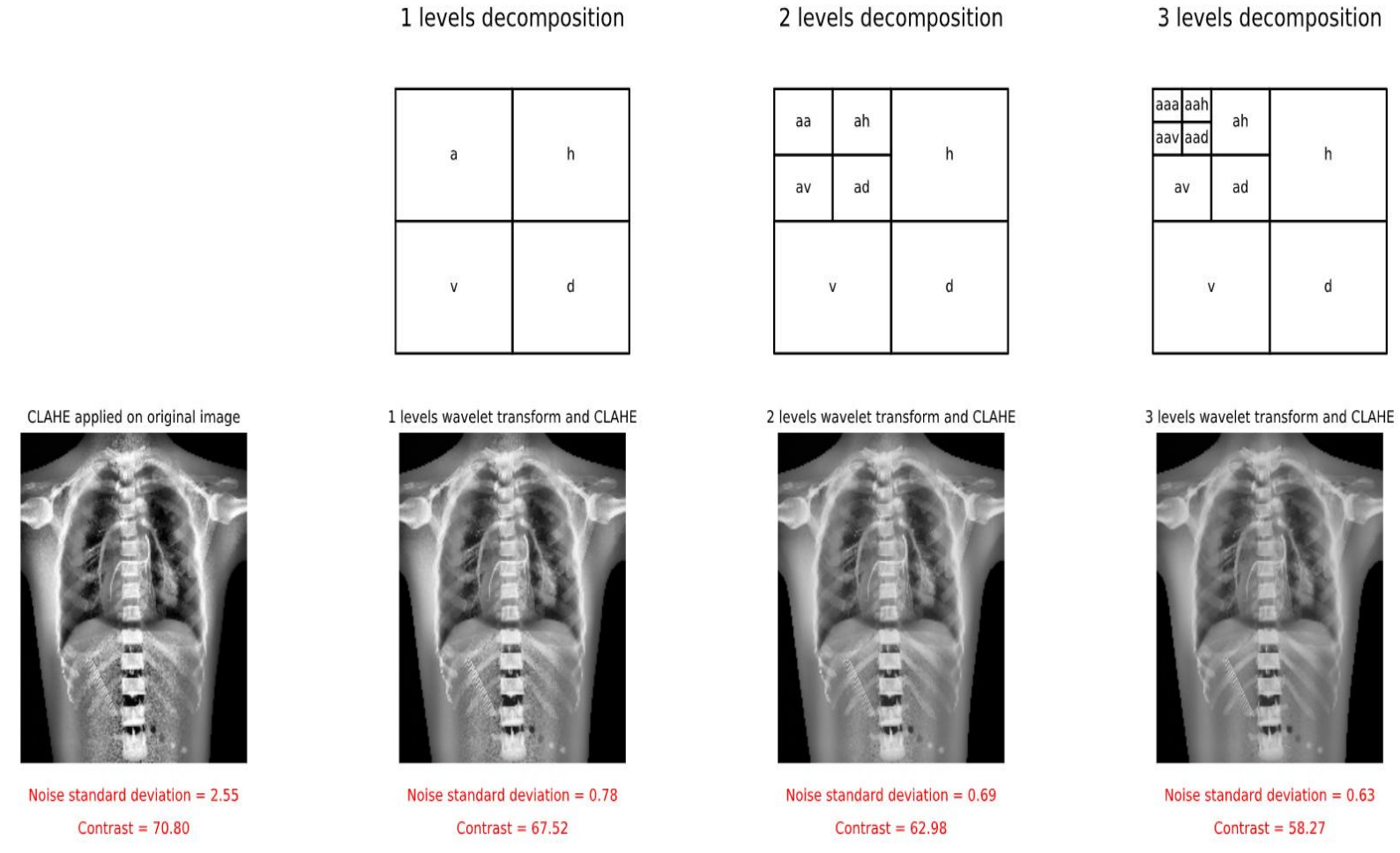
Méthodes multi échelles

Décomposition en ondelette

Contrairement à la transformation de fourrier qui sacrifie complètement la précision temporelle au profit d'une grande précision en fréquence, les ondelettes permettent de garder de l'information dans l'espace temporel. De plus, on peut analyser les différentes fréquences avec des résolutions également différentes ce qui permet une plus grande modularité. Par exemple on n'a pas nécessairement besoin d'avoir une bonne localisation en temps pour les basses fréquences alors qu'on aimerait bien distinguer les hautes fréquences temporellement. La transformé de fourrier approxime un signal par une somme de fonctions sinus, c'est-à-dire une fonction qui oscille sans interruption ce qui n'apporte pas d'information sur le temps, uniquement sur les fréquences. De plus, la fonction sinus étant périodique, elle n'est pas adaptée pour approximer des signaux aux changements abrupts qui sont les types de signaux que l'on retrouve dans les images, notamment aux bords des textures. A l'inverse les ondelettes n'existent que pour une période de temps fini et ont une moyenne de zéro. En translatant une ondelette le long d'un signal (convolution) et en faisant varier son échelle (scaling), on est capable d'approximer les signaux plus erratiques et instables. En effet, plus l'ondelette est large et étirée, mieux on approximera les parties du signal ayant des basses fréquences, c'est-à-dire les parties du signal ayant des variations lentes. A l'inverse, les ondelettes plus étroites permettront d'approximer les hautes fréquences du signal, c'est-à-dire les changements plus prononcés.

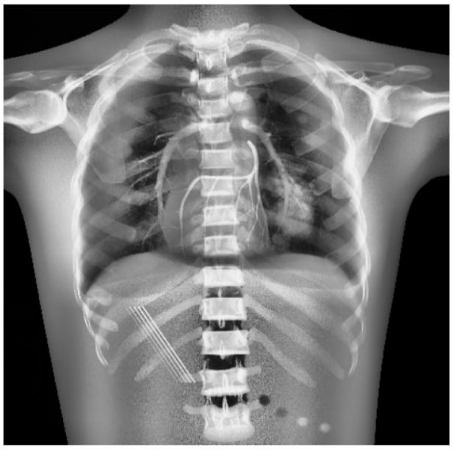
Nous allons utiliser la transformée en ondelette discrète (DWT) de manière à décomposer notre image en hautes et basses fréquences. Dans l'exemple qui suit, nous procédons à une décomposition en trois niveaux. A chaque niveau de décomposition, on décompose à nouveau les basses fréquences en hautes et basses fréquences. A chaque niveau de décomposition, le nombre de coefficients est divisé par quatre (un octave). Les basses fréquences, aussi appelés coefficients d'approximations contiennent le gros de l'information de l'image. Les hautes fréquences, aussi appelés les coefficients de détails représentent les détails de l'image comme les contours par exemple. Ce sont ces hautes fréquences qui sont responsable du bruit dans l'image.

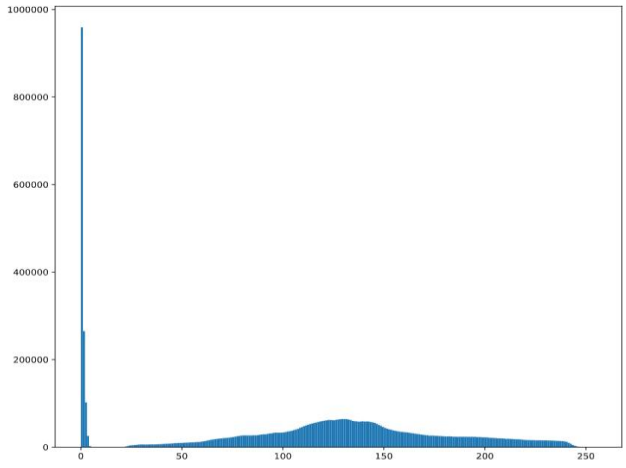
On a vu plus haut que CLAHE avait tendance à amplifier le bruit gaussien dans l'image. Par conséquent, il serait intéressant d'appliquer cet algorithme uniquement sur les basses fréquences qui ne contiennent pas de bruit. A l'aide de la bibliothèque 'pywavelet' on va effectuer cette décomposition puis appliquer CLAHE uniquement aux basses fréquences et ce, pour trois niveaux de décomposition. Par ailleurs, une fois le traitement par ondelettes effectué, on va remplacer les intensités des pixels dans la zone de feu nu par les valeurs des pixels dans l'image originale, c'est à dire des valeurs très faibles, proches de zéro (fond noir donc). De cette manière on aura une estimation plus réaliste du bruit puisque la seule zone qui nous intéresse se trouve au niveau du buste. En effet, sans ce remplacement, on peut constater que beaucoup de bruit est généré dans la zone de feu nu ce qui aurait biaisé les résultats. On a utilisé un masque numpy pour cela. Voici le résultat obtenu avec l'ondelette de debauchies à deux moments :



Sur les schémas de la première ligne a représente les coefficients d'approximations (basses fréquences). V, d et h représentent les coefficients de détails (hautes fréquences) respectivement dans le sens verticale, diagonale et horizontal. La mesure du contraste est effectuée en prenant l'écart type de l'image.

On remarque qu'à mesure qu'on augmente le niveau de décompositions on obtient une image moins bruitée mais aussi avec un taux de contraste plus faible. Cela est logique car plus on descend dans les niveaux de décompositions et plus la quantité de coefficients sur lesquels est appliqué CLAHE est faible. En effet, on applique CLAHE uniquement sur les coefficients d'approximation, c'est-à-dire les coefficients d'approximations purs, au dernier niveau (a, aa, aaa, etc …). Par conséquent CLAHE a une influence de plus en plus faible pour des niveaux de décomposition plus élevés. Le but est donc ici de sélectionner une image avec un bon compromis contraste-bruit. L'image obtenue après un niveau de décomposition semble répondre à ce critère. En effet cette image réduit l'écart type du bruit gaussien par 3.27 tout en conservant un niveau de contraste acceptable.

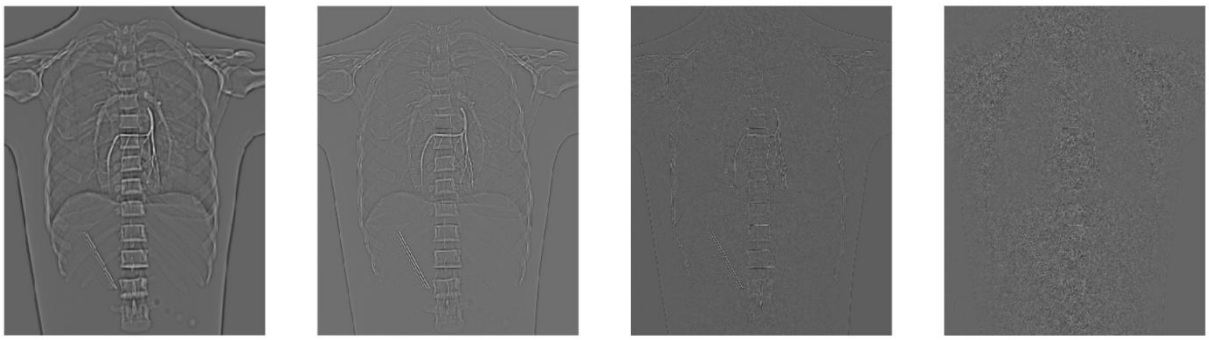




Au final, on constate que l'histogramme de l'image obtenu est plus étiré et couvre une plus large plage de valeurs que l'image d'origine. Seul subsiste un pic de valeurs dans les tons sombres qui représente la zone de feu nu.

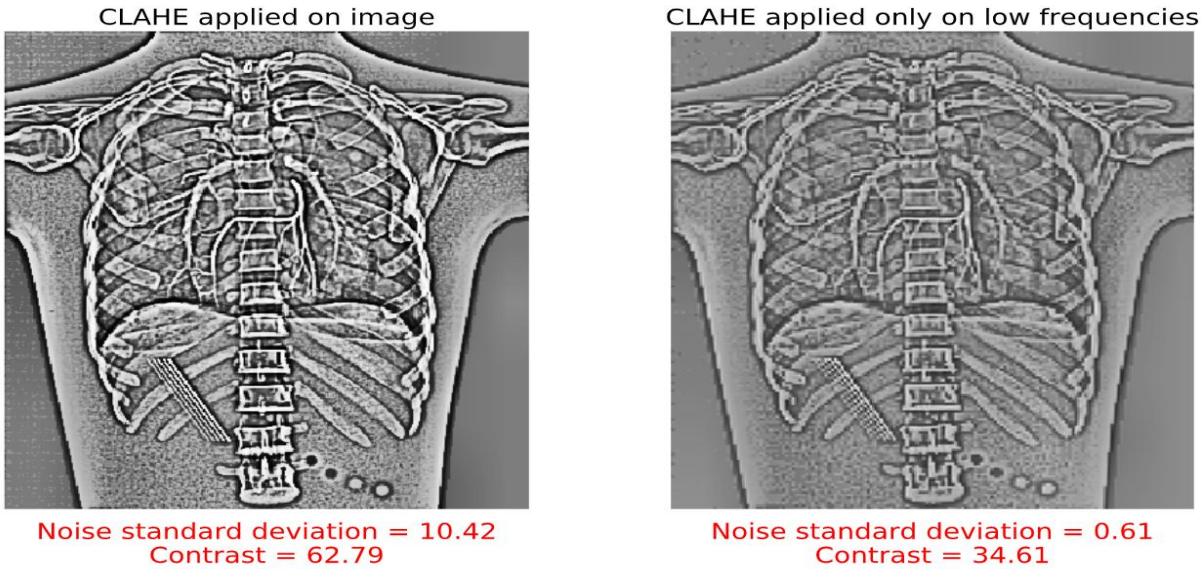
Pyramide laplacienne

En utilisant le concept de pyramide laplacienne, il est possible d'obtenir une image représentant les contours du buste. Cette image serait différente des images exposés jusqu'à présent en ce qu'elle se concentre sur les détails du buste. En effet, la pyramide laplacienne est en fait un filtre passe-haut c'est à dire qu'elle va couper les basses fréquences. Pour construire cette pyramide laplacienne, on effectue, à chaque niveau d'une pyramide Gaussienne (pyramide obtenu en floutant l'image puis en divisant sa taille par 4), la différence entre l'image à ce niveau et l'image du niveau supérieur (qui est une image dont la taille est un quart de celle du niveau inférieur) agrandi pour correspondre au niveau actuel. Pour obtenir les images aux différents niveaux de la pyramide Gaussienne on utilise les fonctions 'pyrdown' et 'pyrup' d'openCV. Voici les images de la pyramide laplacienne sur quatre niveaux :



De gauche vers la droite on a les images du plus haut niveau de la pyramide vers le plus bas. C'est-à-dire des images à plus faible résolution vers les plus hautes. Par exemple l'image la plus à droite correspond à un seul niveau au dessus de l'image originale.

L'image la plus à gauche semble être la plus acceptable. On discerne assez bien les détails des bronches ainsi que des côtes. Néanmoins l'image manque de contraste. On va donc appliquer CLAHE sur cette image. On va également comparer l'image obtenu avec l'application de CLAHE uniquement sur les basses fréquences grâce à la décomposition en ondelettes. Ici on ne décomposera que sur un seul niveau, les résultats seront déjà assez parlants.

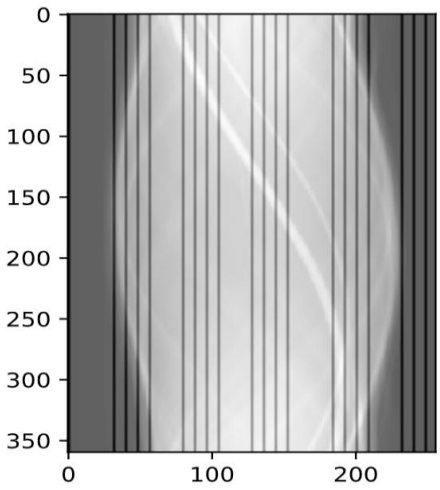


On peut voir que, en appliquant CLAHE uniquement sur les basses fréquences on a considérablement réduit le bruit gaussien. Cependant le contraste a été quasiment divisé par deux. En pratique, il me semble que l'image de gauche, même si elle est très bruitée, reste plus exploitable que l'image de droite. Le contraste étant trop faible sur cette dernière.

TP3 : Comment envisageriez-vous de corriger de tels artefacts liés aux non-idéalités du détecteur ?

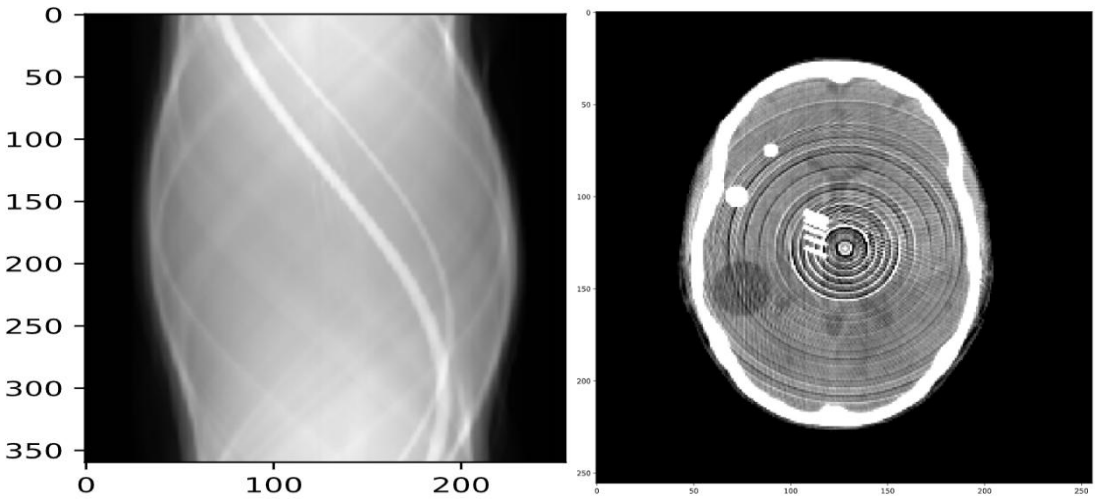
Une première manière d'éliminer les artefacts circulaires présents dans l'image reconstruite serait de simplement regarder le sinogramme. En effet pour des valeurs du gain et de l'offset suffisamment grandes, les artefacts présents dans l'image reconstruite seront également présents dans le sinogramme sous la forme de lignes verticales. Cela permet de localiser la partie du détecteur fautive

pour mieux la calibrer. Ci-dessous, pour un gain suivant une loi normale de moyenne dix on voit très clairement apparaitre ces lignes :

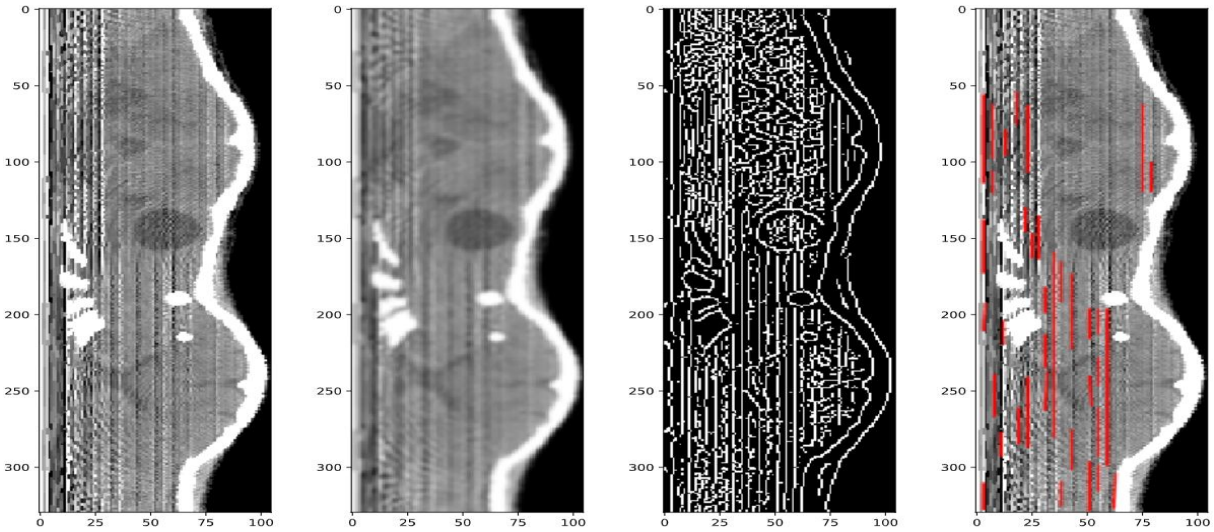


On remarque que, par exemple, l'élément 103 du détecteur (à peu près) à un problème et que ce problème se manifeste naturellement pour tous les angles de mesure représentés par l'axe des ordonnées. Cela explique la présence de ces lignes noires.

Cependant, pour des valeurs de gain et d'offset plus faible, par exemple suivant une loi normale de moyenne 0, ces lignes deviennent difficilement repérables. Par conséquent il devient nécessaire de repasser sur l'image reconstruite où les artefacts sont plus visibles.



Néanmoins, il semble assez difficile de détecter les cercles dans l'image. Il serait plus simple de détecter des lignes pour pouvoir les corriger ensuite. Pour transformer les cercles en lignes on peut effectuer une transformation en coordonnées polaires de l'image. On peut utiliser la fonction warp_polar de skimage ou warpPolar d'opencv pour obtenir une représentation en coordonnées polaires de l'image.



Ci-dessus on a à gauche l'image en coordonnées polaires. Deuxième image, cette même image à laquelle on a appliqué un filtre Gaussien de manière à réduire le bruit. Dans la troisième image, on a appliqué un filtre de Canny de manière à détecter les bords dans l'image. A droite on a essayé de détecter les lignes avec la fonction houghLineP d'opencv. On remarque que les lignes détectées en rouge sont morcelées et que toutes les lignes n'ont pas été détectées. Si on avait réussi à détecter ces lignes on aurait pu appliquer une correction puis effectuer la transformée inverse vers les coordonnées cartésiennes pour récupérer notre image de cerveau sans les artefacts. Ci-dessous on a notre image reconstruite en gardant les lignes détectées en rouge, donc sans traitement :