

tp irm enseirb : kspace

Aurélien Trottier, Valéry Ozenne

March 14, 2019

Partie 1: Transformée de Fourier

Charger le jeu de donnée `kspace_brain_1ch.mat` situé dans le répertoire `data` avec la fonction `load(filename)`.

Vous disposez maintenant d'une matrice complexe de dimensions 160x220 voxels, avant d'effectuer la reconstruction IRM, observons les données.

Tracer le profil en intensité (valeur absolue) de la ligne centrale dans la première direction puis seconde direction, vous pouvez utiliser les commandes suivantes:

`figure, subplot, plot,`

Tracer ensuite les mêmes profils en intensité après application de la fonction `fftshift`

Tracer ensuite les profils en intensité après application de la transformée de Fourier inverse avec la commande suivante:

`ifft`

Afficher le kspace, seuiller l'image en prenant comme valeur maximale d'affichage 5% de la valeur maximum du kspace

Reconstruire l'image à partir des données complexes et afficher la partie réelle, la partie imaginaire. Que voyez-vous ?

`ifft2, fftshift (2x) , real, imag`

Calculer l'image de module et l'image de phase $module = \sqrt{Reel^2 + Imag^2}$ et $phase = \arctan2(Imag/Reel)$, comparer le résultat avec les fonctions suivantes:

`abs, angle`

Effectuer la transformée de Fourier inverse de la ligne centrale dans la dimensions 1. L'intensité obtenue correspond à la somme du signal acquis perpendiculairement. Vérifier cette hypothèse en sommant l'intensité obtenue dans la direction perpendiculaire.

Retroprojection en IRM : jeter un coup d'oeil ici:

<http://irmcardiaque.com/index.php?title=Fourier3>

Partie 2 : Modification du kspace et Partial Fourier

Définir `dimx` et `dimy` comme dimension du kspace

Enlever le centre du kspace sur un carré de 10 voxels et effectuer la reconstruction. Que notez-vous?

Enlever les bords du kspace en retirant 10 lignes ou colonnes de voxels et effectuer la reconstruction. Que notez-vous?

Enlever la partie inférieure du kspace en retirant 10 lignes de voxels. Que notez-vous ?
Afficher ces 3 résultats sur une unique figure avec l'image de référence.

Partie 3 : Modification du kspace et Zerofilling

Copier le kspace dans une matrice 2x fois plus grande en le disposant au centre et effectuer la reconstruction? Que voyez-vous ?

Rappelez le lien entre la taille du kspace, le FOV et la résolution de l'image.

Ajouter aléatoirement dans le kspace quelques points proche de valeur maximale du kspace. Effectuer la reconstruction, que voyez vous ?

Partie 4 : Combinaison des antennes

Charger le jeu de donnée `brain_8ch.mat` situé dans le répertoire `data` avec la fonction `load(filename)`.

Vous disposez maintenant d'une matrice complexe de dimensions 160x220x8 voxels. La 3^{ème} direction correspond aux nombres d'éléments en réception (antennes).

Afficher pour chaque antenne, le module de l'image. Que notez-vous ?

Combiner les éléments en sommant les images complexes ? Que notez-vous?

Calculer la transformée de fourier de l'image complexe sur chaque antenne pour revenir dans l'espace réciproque (kspace). Combiner les éléments en sommant les kspaces complexes . Revenir dans l'espace image, que notez-vous?

Enfin combiner les éléments avec la méthode du "sum of square" sur les images complexes?

Calculer la transformée de fourier sur l'image complexe combinée. Nous utiliserons ces données pour la suite de cette partie.

Nous allons ensuite modifier le kspace pour visualiser l'impact sur l'image.

Ajouter un point ayant pour valeur 200 aux voxels de coordonnées (70,60). Effectuer la reconstruction.

Ajouter un point ayant pour valeur 200 aux voxels de coordonnées (70,130). Effectuer la reconstruction.

Partie 5 : Effet d'un gradient sur l'image

Nous allons simuler l'effet d'un gradient sur le mouvement dans l'image. Pour cela nous allons introduire un décalage en phase dans la seconde direction (la direction d'encodage de phase). Le décalage introduit est le même partout selon la première direction (readout), il n'y a pas de variation introduite en readout.

Choisir un champ de vue (FOV) de 200 et on offset de 30. Appliquer le gradient suivant :

$$kspace - shift(x, y) = kspace(x, y) * gradient(y)$$

avec $gradient(y) = \exp((y - dimy/2 + 1)/sy * dimy * offset/FOV)$

Faire varier l'offset.