tp irm enseirb : sense

Aurélien Trottier, Valéry Ozenne

March 14, 2019

Partie 1 : Les cartes de sensibilité

Charger le jeu de donnée `kspace_brain.mat` situé dans le répertoire `data`. Vous disposez maintenant de deux matrices complexes `image_brain` (l'image reconstruite complexe) et `smaps` (la carte de sensibilité des antennes) de dimensions respectives: 256x256 et 128x128x8. Pour que les deux matrices aient les mêmes dimensions , nous allons diminuer la résolution de `image_brain` d'un facteur deux avec la fonction `resize`. Afficher `image_brain` et `smaps`.

Pour `smaps`, vous pouvez utiliser la fonction suivante `ismrm_imshow`:

```
ismrm_imshow(abs(smaps),[0 max(abs(smaps(:)))],[2 4]);
```

Ensuite nous allons créer synthétiquement les images acquises en absence de bruit. Pour cela utiliser la fonction `ismrm_sample_data` avec une accélération de 1, seulement les 3 premiers arguments de cette fonction sont nécessaires. Afficher les images pour chaque antenne.

Pour rappel :

$$Img_{acquise} = Img_{réelle}\ S + B$$

avec S la carte de sensibilité des antennes et B le bruit.

Partie 2 : Le sous échantillonnage du kspace

Pour gagner du temps à l'acquisition, il est possible de sous échantillonner le kspace. C'est à dire que l'on va acquérir moins de lignes de l'espace de Fourier, par exemple uniquement un ligne sur deux (accélération = 2).

Simuler cette acquisition en effectuant la reconstruction en enlevant sur le kspace une ligne sur deux pour chaque antenne. Vous aurez ainsi une matrice 256x128x8, , noter la `img_test`.

Effectuant la reconstruction en mettant à 0 une ligne sur deux du kspace, cette fois ci votre matrice reste de dimension 256x256x8, noter la `img_alias`.

Afficher les résultats avec `ismrm_imshow`.

Partie 3 : SENSE sur un voxel.

Nous allons maintenant voir comment il est possible de récupérer l'information manquante en exploitant la sensibilité des antennes et la présence d'information redondante entre les canaux.

Choisir dans la dimension 1 une ligne du kspace par ex x=128;

Extraire pour cette ligne les données correspondantes de `img_alias` et de `smaps`. Vérifier que `ligne_alias` et `ligne_smaps` soir bien de dimensions 128x8.

Choisir un voxel dans la ligne par ex x=32 et trouver le pixel aliasé correspondant.

Pour rappel, le problème est le suivant:

$$Img_{aliasé} = Img\ S$$

avec S la carte de sensibilité des antennes

$$Img = (S'S)^{-1}S\ Img_{aliasé}$$

$$Img = Umix\ Img_{aliasé}$$

where $Umix = (S'S)^{-1}S$

Calculer Umix sachant que S correspond à la matrice des cartes de sensibilité des 2 voxels aliasés, S doit être de dimension `8x2`. Vous pourrez utiliser la fonction `pinv` pour effectuer l'inversion

Partie 4 : SENSE sur toute l'image

Généraliser le calcul sur chaque voxel de l'image puis sur chaque ligne de l'image afin d'obtenir la carte de unmixing de dimension 256x256x8. Les afficher.Reconstruire l'image finale en combinant les antennes.

Partie 5 : Influence du bruit

Refaire l'étape 4 en ayant préalablement ajouter du bruit sur le kspace. Effectuer la reconstruction avec une accélération de 2 puis de 4 pour observer les différences. Quelle est l'influence de l'accélération en présence de bruit ?