MRI Recon Lab

FT and SENSE

November 20, 2023

1 Cartesian Undersampling

Provided Material

- **M.mat** 2D image data
- **itok.m** This function transforms data from image domain to k-space using the Fast Fourier Transform (FFT). Example: KSpaceData = itok(ImageData);.
- **ktoi.m** This function transforms data from k-space to image domain using the inverse Fast Fourier Transform (iFFT). Example: ImageData = ktoi(KSpaceData);.

Assignment

(a) Simulate an MR data acquisition using the function itok.m to transfer the image data M to the k-space domain yielding the k-space data K. Depict the image and k-space data K. Hint: Remember that M and K should be complex data. Display log(abs(K)) rather than abs(K) for better visualisation

(b) Simulate a Cartesian sampling pattern with a uniform undersampling factor of 2 in one direction by creating a zero matrix U_2 with the same size as K and then set every second horizontal line to 1. Following a similar procedure simulate Cartesian sampling patterns with a uniform undersampling factor of 5 (U_5) and with a random undersampling factor of 3 (U_{R3}) in one direction. Depict the undersampling patterns and compare them.

(c) Apply the undersampling patterns by multiplying K with the corresponding undersampling pattern U and transform the obtained k-space data to image space using ktoi.m yielding M_U. Depict and compare the undersampled images. How would you expect the undersampled images to look like if the undersampling would have been performed in the other direction?

(d) A commonly used approach to study the effect of different sampling patterns on the final image quality is to analyze the point spread function *PSF* corresponding to each sampling pattern. The *PSF* describes the response of an imaging system to a single delta function. A delta function in image space is equivalent to a constant k-space signal. Therefore, the PSF for each undersampling pattern can be obtained by multiplying the corresponding sampling pattern U with a constant 2D k-space (i.e. `KPSF = ones(N,N);`) and then transforming it to image space. Depict the PSF for each of the Cartesian undersampling patterns (U_2, U_5 and U_{R3}) and compare them. What happened to the delta peak input for these sampling patterns?

2 Iterative SENSE

Provided Material

- **C.mat** 2D coil sensitivity information for eight coils.
- **M.mat** 2D fully sampled MR image data.
- **ktoi.m** This function transforms data from k-space to image space using the inverse Fast Fourier Transform (iFFT). Example: `ImageData = ktoi(KSpaceData);`.
- **itok.m** This function transforms data from image space to k-space using the Fast Fourier Transform (FFT). Example: `KSpaceData = itok(ImageData);`.

Assignment

In this exercise we will reconstruct Cartesian undersampled data using iterative SENSE. Consider the fully sampled image m.mat and the coil sensitivity maps C.mat from an 8-channel brain coil (Nc = 8). Apply the coil sensitivity information to the image data creating the coil images m_c obtained by each of the coils.

(a) Determine and depict m, $C_i, i = 1,2,...,N_c$ and $m_i, i = 1,2,...,N_c$. What can you say about the configuration of the coils?

(b) Employ the root-sum-of-square approach and the weighted coil sensitivity approach to combine the data from the individual coils. Depict and compare the combined images with respect to the original fully sample image m.

(c) Generate 2 uniform undersampling patterns with acceleration factors of 4 and 8 (U_4 and U_8) and 2 random undersampling patterns (U_{R4} and U_{R8}). Undersampling should be performed along the horizontal direction. Each sampling pattern must be a matrix with 1s in the sampled positions and 0s in the remaining entries.

(d) Obtain the aliased images for each coil as a result of undersampling with the generated patterns. For this you should use:

$$b_i = F^{-1}UFC_im$$

where U is the corresponding undersampling pattern, F is the Fourier transform and b_i are the aliased images for each coil $i = 1,...,N_c$. Depict and compare the aliased images for the different undersampling factors and patterns.

(e) The SENSE undersampled reconstruction can be written as a linear problem:

$$E\hat{m} = b$$

where $\hat{m}$ is the image to be reconstructed and the encoding matrix $E = UFC$ corresponds to the forward sampling operator, similar to what was described in d), with U the undersampling operator, F the Fourier transform operator, C the coil sensitivity maps and b the undersampled k-space data. In the folder */@Cartesian SENSE* you will find the implementation of the operator E as a MATLAB class. An instance of this operator can be created via:

```
E = Cartesian_SENSE(U,C) ; % create class instance
```

Defining the operator as MATLAB class enables us to easily overload the multiplication operator. Thus, we can apply the operator E like this:

```
b = E * M; % forward operator
m = E'* b; % conjugate transpose operator
```

The forward operator E represents the k-space acquisition (i.e. going from image space to k-space). The conjugate transpose operator E^H represents the image reconstruction (i.e. going from k-space to image space). The file *mtimes.m* overloads the multiplication operation. You can look into this function to see how it works.

To test this class, generate a uniform undersampling pattern of 4 U_4 and create an instance of the corresponding encoding operator. Simulate the acquisition on image M with the forward operator to obtain b. Following, perform a zero-fill reconstruction using the conjugate transpose operator to obtain m. Depict and comment on the zero-fill reconstruction.

(f) Iterative SENSE reconstruction is obtained by solving the linear problem $E\hat{m} = b$ as a least square minimization $min_m||Em - b||_2^2$. Implement the Gradient Descent method to solve the equivalent normal equation:

$$min_m||(E^H E)m - E^H b||_2^2$$

To do this, you can use the forward E and conjugate transpose E^H encoding operators you implemented in e). Your function for the Gradient Descent should be in the following syntax:

```
function   rho = Gradient_Descent (b,E, maxit)
```

where b is the k-space data, E is the encoding operator and *maxit* is the maximum number of iterations. You can use the following Gradient Descent algorithm:

$$m_0 = E^H b$$

$$r_0 = m_0 - E^H E m_0$$

$$\alpha_k = \frac{r_k^T r_k}{r_k^T E^H E r_k}$$

$$m_{k+1} = m_k + \alpha_k r_k$$

$$r_{k+1} = r_k - \alpha_k E^H E r_k$$

Show and compare your results for the undersampling patterns generated in question c). What can you conclude from them? How many iterations are needed to reconstruct acquisitions with the different sampling patterns?