Computational MR imaging
Laboratory 2: k-space sampling and Fourier reconstruction

Report is due on Wednesday before the next lab session at 23:50. Please upload your report on StudOn.

Learning objectives

- Reconstruct Cartesian MRI data
- Compute the point spread function
- Apply zero-padding and filtering in k-space

1. FFT reconstruction of Cartesian MRI data

- Load the file kdata1.mat
- Reconstruct kdata1 using FFT (create functions fft2c and ifft2c to perform forward and inverse Fourier transforms for MRI data as described in class, see numpy.fft Python module)
- Plot the magnitude and phase of the reconstructed image

2. Effects of k-space zero-padding and zero padding

- Truncate kdata1 by taking the central 64×64, 128×128, and 256×256 k-space points
- Zero-pad to 512×512 and reconstruct the corresponding images. Plot the magnitude of the reconstructed images and their difference with respect to the original reconstruction from part 1
- Discuss the effects of k-space truncation

3. Point spread function (PSF)

- Details about PSF will be explained in the exercise.
- Compute the (1D) PSF along the x dimension for each k-space truncation pattern in part 2
- Plot the different PSF's in the same figure using different colors
- Compute the full-width at half-maximum (FWHM) for each PSF
- Discuss the differences in spatial resolution

4. k-space filtering (windowing)

- Multiply each truncated k-space data with a Hamming window (see the method "window" in the Python module, "skimage.filters"), zero-pad to 512×512 and perform an FFT reconstruction
- Compute the PSF along the x-dimension for each window and compare it to the PSF from part 3, where a rectangular window was implicitly used for the reconstruction
- Compute and compare the FWHM of the PSF of the Hamming window and again compare it to the rectangular window from part 3
- Discuss advantages and disadvantages of the different windows with respect to each other, in particular the effects on resolution and Gibbs ringing. Hint: Take a look at the sidelobes of the PSF

5. Oversampling the readout dimension

Oversampling along the frequency-encoding dimension (readout) is usually employed to avoid wrap-around artifacts.

– Load the file kdata2.mat. The k-space data has dimensions [168,336]. The frequency-encoding dimension (second dimension) was oversampled by a factor of 2. Remove the oversampling by decimating the frequency-encoding dimension. Reconstruct the oversampled and decimated data sets. Discuss the effect of oversampling.