Project 1

Alternative 1: Compressed sensing in MRI

Introduction:

One property of sequences for acquiring MR images is their acquisition time. Several arguments exist for preferring a low acquisition time which depend on the particular application case. Since the acquisition of the full k-space assumes that the measured object of interest is static, patient movement may create artefacts in the image after reconstruction. For that reason the acquisition time is generally desired to be as low as possible. Apart from static images, other potential applications like angiography, i.e. measuring time-resolved data, benefit from a shorter measurement time for each image frame as well.

As we know from the lecture, the sampling of k-space has an impact on the resulting image data and is therefore restricted by Nyquist's theorem. However, there are different ways to employ additional pre-knowledge on the measured data that enables one to apply undersampling when acquiring k-space data while still being able to reconstruct reasonable images.

A famous such technique is based on *Compressed Sensing* (CS) which was proposed to be used in the context of faster MR acquisitions in a paper by Michael Lustig in 2007 [1] in a quite descriptive way. This will be the basis for this project.

During the project you will work on the implementation of a simple approach for CS reconstruction from simulated, randomly undersampled k-space data in Python. While the main code for the intended algorithm is already available in Matlab [2], you will further conduct small experiments in order to investigate the purpose of the main idea in this context, i.e. *random undersampling*.

Tasks:

1. Get an easy start by reading the entertaining comic at provided at [2]. After that, skim through the two papers provided at [2] and try to **summarize any information** you find in them **on the motivation for using CS in MRI**.
2. Have a look at the section *Workshop* at [2]. Download the **code** package and run the four demo files in Matlab. Use the complementary **slides** for further explanations. Find out **which are the particular steps** composing the reconstruction algorithm in *demo4.m*. In the report, present these steps using **a suitable diagram**.
3. Have a look at Figures 4 and 5 in [1]. Explain the **concept of *TPSF*** (see Eq. 2). What are the **desired properties for the TPSF**? Provide **Python code** to **reproduce the experiments** in the spirit of Figures 4 and 5.
4. Provide **Python code** to perform the experiments in *demo4.m*. Investigate the **effect of** using **different sampling patterns** in k-space, i.e. vary used *pdf*, as well as **modifying the thresholding approach**.
5. Based on your experiences from Tasks 1-4, what are the ***key ingredients*** (or main principles) **of** the **Compressed Sensing** algorithm.

Assessment:

The project work can be done in groups of **2 people**. Each group has to hand in a report of **maximum 4 pages** describing

1. underlying **problems**
2. basic **theories**
3. solution **strategies** used
4. findings from the **experiments**
5. final **conclusions**.

Make sure to **include contents for each** of the given **task**s. The highlighted parts may help to put emphasis on certain aspects.

Each group will do a **demo and present** its **results** to the rest of class the **16th of May**. Each group will have **7 min** for this. The report and code must be submitted not later than the **14th of May**. The grades of the mini-exams of the modules *Reconstruction* and *Restoration* are part of the final assessment for this project:

Report:	40 points
Demo & presentation:	20 points
Code:	20 points
Mini exams:	20 points

Good luck!!!

References:

[1] "Sparse MRI: The application of compressed sensing for rapid MR imaging" by M Lustig et al. MRM 2007 http://dx.doi.org/10.1002/mrm.21391

[2] Section *Compressed Sensing MRI* at personal page by M Lustig, http://people.eecs.berkeley.edu/~mlustig/CS.html