Nuts & Bolts of Advanced Imaging – The Image Reconstruction Pipeline

Michael S. Hansen (michael.hansen@nih.gov)

Highlights:

- The role of the image reconstruction process
- Typical image reconstruction components
- Implementation of modular reconstruction components
- Existing (commercial and non-commercial) reconstruction pipeline systems

TARGET AUDIENCE: Anybody with an interest in MRI reconstruction software beyond theoretical principles and who would like the tools to implement practical, production-level imaging reconstruction software.

OBJECTIVES: This course aims to develop a description of the image reconstruction process as a pipeline processing system. Furthermore, the aim is to describe each of the steps in the pipeline processing chain as modular entities that can be modified and replaced somewhat independently of each other. The benefits of such a modularized pipeline will be illustrated and example reconstruction systems will be discussed.

PURPOSE: Magnetic Resonance Imaging (MRI) data is not acquired in image space. A set of processing steps is required to transform the acquired raw data into images that can be displayed and interpreted by a clinician. This transformation of raw data into images is known as image reconstruction [1]. The image reconstruction is performed by dedicated software, which on most clinical MRI systems is running on dedicated reconstruction computer hardware. The purpose of this course is to break down this reconstruction process into a set of common processing steps that are repeated in most reconstructions. The function and implementation of these basic modules will be discussed. In modern reconstruction systems the processing steps are modular such that they can be reused in multiple reconstruction programs where they are usually assembled in a pipeline processing architecture. This pipeline processing architecture can be found in most vendor reconstruction systems and in open source systems that will be discussed in this course.

METHODS: The role of the image reconstruction system is outlined in Figure 1. The magnet with gradients, RF hardware, and spectrometer forms the image encoding part of the MRI system. The encoded MRI signal is sampled in the spatial frequency domain (k-space) and the image reconstruction process is responsible for transforming this signal to image space. Conceptually, this image reconstruction process is just an inverse Fourier transform, but modern reconstruction systems contain several other steps that are crucial for proper image reconstruction. In this course we use 2D Cartesian parallel imaging reconstruction [2–4] as an example and explore various ways this can be implemented in a pipeline architecture.

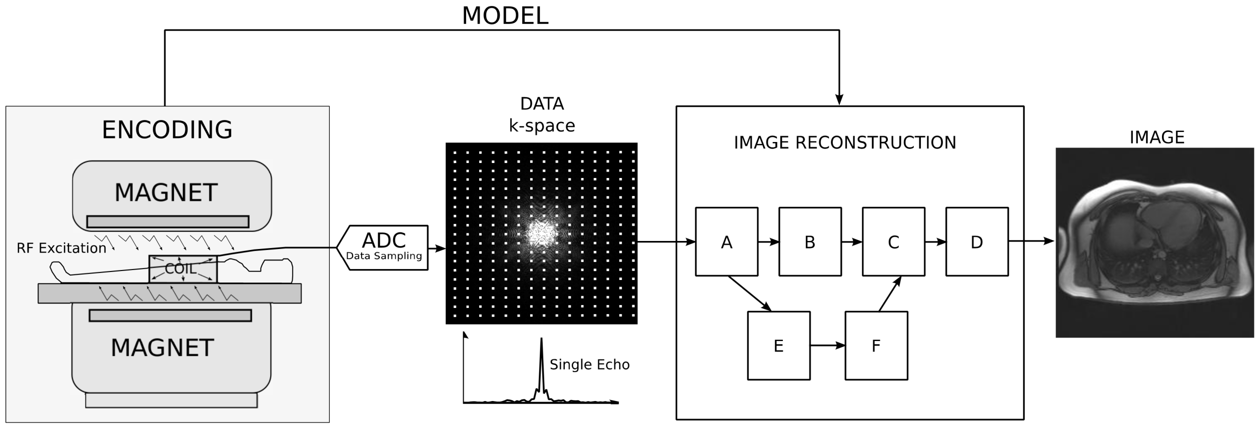

Figure 1. The role of the reconstruction process in the MRI measurement. The MRI hardware with radiofrequency transmit/receive system and spatial encoding gradients serves as the image-encoding device. The actual measurement produces encoded imaging data (k-space), which is transformed by the image reconstruction process into images. The task of the image reconstruction process is to use information (a model) about the performed encoding steps to transform the data into images. The reconstruction process consists of multiple signal processing steps that are frequently depicted as steps in a reconstruction pipeline; here the steps are enumerated A–F.

A typical image reconstruction pipeline for parallel imaging is shown in Figure 2. This reconstruction pipeline includes the modular steps a) noise pre-whitening [5], b) k-space filtering, c) zero-filling, d) Fourier transform, e) parallel imaging calibration [2–4], and f) coil combination [6, 7]. Most of these steps will be shared among several reconstruction pipelines but some may be modified or replaced by others. For example, the raw data filtering in k-space may be modified based on the specific needs of the application (balancing spatial resolution and Gibbs ringing). Noise adjust is also an optional step that can be left out, albeit at the expense of some loss in SNR.

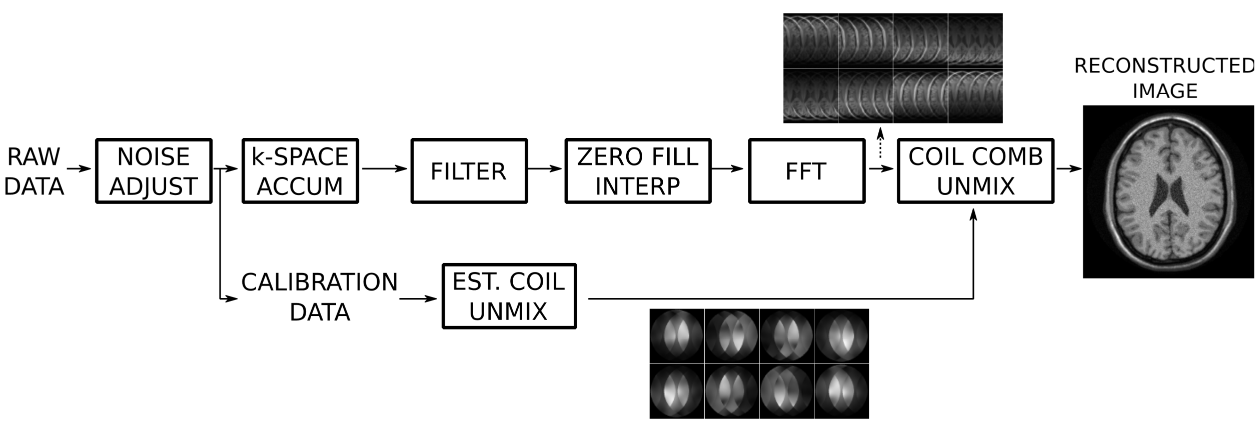

Figure 2. Illustration of a typical reconstruction pipeline for a Cartesian parallel imaging acquisition. The first step in the reconstruction is noise prewhitening (noise adjust), which removes noise correlation in the data. The data pipeline then splits into two, one for the main image reconstruction and one for the processing of calibration data to form parallel imaging unmixing coefficients. The main processing pipeline performs raw data filtering, zero filling in **k** space to ensure square pixels (image interpolation), Fourier transform, and finally coil combination using the parallel imaging unmixing coefficients. This final step turns the aliased channel images into a single combined image.

In this lecture, we will review the implementation of the reconstruction steps using commonly used scripting languages, e.g. Matlab or Python and we will provide example implementations for the participants to review and modify. Emphasis will be put on generality of the chosen implementation strategies. Specifically, we will discuss how parallel imaging calibration (whether a SENSE or GRAPPA type approach is chosen) can lead to a set of coil combining coefficients that can be used to combine a set of aliased coil images into an unaliased image. The role of scaling in the individual steps and tools for predicting the noise characteristics of the reconstructed images will also be discussed. The presented tools will operate on data stored in ISMRMRD format (http://ismrmrd.github.io).

Following this discussion of the algorithmic steps involved in each module, we will review how these algorithms can be integrated into existing pipeline architectures. Commercial pipeline processing tools are usually available from the MRI manufacturers and some manufacturers provide useful programming windows through research agreements. Alternatively reconstruction pipelines can be developed using Open Source tools such as the Gadgetron [8] or the Graphical Programming Interface (GPI) [9]

CONCLUSION: The purpose of the image reconstruction software is to transform the k-space raw data to images. An image reconstruction program can be divided into logical steps that can be modified or exchanged independently of each other. This lecture demonstrates how commonly used image reconstruction building blocks are implemented and how they can be integrated into scripting environments for prototyping or actual high-performance reconstruction pipelines.

REFERENCES:

1. Hansen MS, Kellman P: **Image reconstruction: An overview for clinicians.** *J Magn Reson Imaging* 2014.

2. Sodickson DK, Manning WJ: **Simultaneous acquisition of spatial harmonics (SMASH): fast imaging with radiofrequency coil arrays.** *Magn Reson Med* 1997, **38**:591–603.

3. Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P: **SENSE: sensitivity encoding for fast MRI.** *Magn Reson Med* 1999, **42**:952–62.

4. Griswold MA, Jakob PM, Heidemann RM, Nittka M, Jellus V, Wang J, Kiefer B, Haase A: **Generalized autocalibrating partially parallel acquisitions (GRAPPA).** *Magn Reson Med* 2002, **47**:1202–10.

5. Kellman P, McVeigh ER: **Image reconstruction in SNR units: a general method for SNR measurement.** *Magn Reson Med* 2005, **54**:1439–47.

6. Roemer PB, Edelstein WA, Hayes CE, Souza SP, Mueller OM: **The NMR phased array.** *Magn Reson Med* 1990, **16**:192–225.

7. Walsh DO, Gmitro AF, Marcellin MW: **Adaptive reconstruction of phased array MR imagery.** *Magn Reson Med* 2000, **43**:682–90.

8. Hansen MS, Sørensen TS: **Gadgetron: An open source framework for medical image reconstruction.** *Magn Reson Med* 2012.

9. Zwart NR, Pipe JG: **Graphical programming interface: A development environment for MRI methods**. *Magn Reson Med* 2014.