Georg-August-Universität
Göttingen

An alternative approach to the image reconstruction for parallel data acquisition in MRI

Frank Bauer and Stephan Kannengiesser

Nr. 2006-25

Preprint-Serie des
Instituts für Numerische und Angewandte Mathematik
Lotzestr. 16-18
D - 37083 Göttingen

An Alternative Approach to the Image Reconstruction for Parallel Data Acquisition in MRI

Frank Bauer[1], Stephan Kannengiesser[2]

[1]: Institute for Mathematical Stochastics
University of Göttingen
Maschmühlenweg 8-10
37073 Göttingen
Germany
Email: bauer@mathematik.uni-kl.de

[2]: Siemens Medical Solutions
Karl-Schall-Str. 6
91052 Erlangen
Germany
Email: stephan.kannengiesser@siemens.com

Abstract

Magnetic Resonance Imaging with parallel data acquisition requires algorithms for reconstructing the patient's image from a small number of measured k-space lines.

In contrast to well-known algorithms like SENSE and GRAPPA and its flavors we consider the problem as a non-linear inverse problem. Fast computation algorithms for the necessary Fréchet derivative and reconstruction algorithms are given.

1 Introduction

Magnetic resonance imaging (MR imaging, MRI) routinely relies on critical sampling in the three-dimensional spatial frequency space (Fourier domain, k-space) for spatial encoding. A fairly recent development allows to replace some of the time-consuming sequential steps of phase encoding by switched magnetic gradient fields, so-called k-space lines, by parallel acquisition with an array of detector elements with a manifold of non-uniform spatial sensitivities. This partially parallel imaging (PPI) has been proposed in many forms and colorful acronyms such as SMASH [1], SENSE [2], GRAPPA [3], SPACE-RIP [4], etc. In short, PPI allows to reconstruct images from k-space data sets with limited support.

All the known methods have in common that they work in two distinct steps: a calibration step, in which a separately acquired data set, or part of the undersampled data set is used to extract information about or related to the sensitivity characteristics of the detector array, and a reconstruction step, in which the undersampled data set is either completed, or an artifact-free image is synthesized from the undersampled data and the sensitivity information.

A popular concept in PPI is called autocalibration, and works by acquisition of a regularly undersampled k-space grid plus a few k-space lines near the center of k-space, i.e., in the low spatial frequency range.

It should be noted that, although the sensitivity characteristics of the array detector elements are unknown and vary with the patient specific detector placement and choice of diagnosis protocol, the noise characteristics of the detector array can easily be measured with high accuracy.

There is today no method available which performs the image reconstruction in a single process which takes into account both the undersampled imaging data and the extra imaging data or the extra calibration data. Also, there is today no optimality criterion for the quality of the image reconstruction, except for those methods which assume perfect knowledge of the spatial sensitivity profiles of the array detector elements, which is of course not achievable in practice.

In this paper we will show how one can reconstruct both the image and the sensitivity out of the data simultaneously. To this end we will reinterpret this problem as a non-linear inverse problem, see e.g. [5, 6] and the references therein. These have received great attention in the recent times in all areas of non-destructive testing. A specific property of inverse problems is their intrinsic instability which is also an explanation for many effects observed in the practice of PPI. The advantage of the methods which we will introduce here is that we can actually guarantee the optimal order of accuracy in our solutions. In general and without additional information it is not possible to beat these methods apart from a constant factor.

The paper is organized as follows. First we describe the specific model we use and introduce notation. In the second part we will give an explicit calculation of the Fréchet derivative which is an essential part of the later considerations. We will give some considerations to reduce the number of necessary operations. Afterwards we will present different possibilities to solve the non-linear inverse problem including some explicit algorithmic parts. The last sections presents a short numerical experiment using the presented algorithms.

2 The Model

In the following we will describe the model we consider for the whole paper. As we can just measure at discrete positions we will not consider a continuous model right now. However most considerations would carry over to an infinite dimensional setting.

2.1 Fourier Transforms

Later on we will largely need two dimensional Fourier transforms. We will denote the fast Fourier transform by FFT and the inverse fast Fourier transform by $\overline{\text{FFT}}$. The Fourier transform roughly costs the number of elements inverted times its logarithm.

Many quantities will have later on a matrix and a vector interpretation. Whenever we apply the Fourier transform we automatically assume that this is applied on the matrix interpretation.

2.2 Matrix Multiplications

We will need two forms of matrix multiplications. The standard one will be denoted by no point. A second kind of multiplication will be denoted by ×. It holds

$$\{a_{ik}\}_{1\leq i\leq n, 1\leq k\leq m} \times \{b_{ik}\}_{1\leq i\leq n, 1\leq k\leq m} = \{a_{ik}b_{ik}\}_{1\leq i\leq n, 1\leq k\leq m}$$

Computing this quantity costs $n \cdot m$ Operations.

2.3 Image

We want to reconstruct an image. The image $\mathcal{P}$ is considered to be a function

$$\mathcal{P} : \mathbb{B} = \{1, \ldots, \mathcal{P}_{\text{hor}}\} \times \{1, \ldots, \mathcal{P}_{\text{ver}}\} \to \mathbb{C}$$

In this discrete case there is a canonical presentation of $\mathcal{P}$ as member of the matrix space $\mathbb{C}^{\mathcal{P}_{\text{hor}} \times \mathcal{P}_{\text{ver}}} = \mathbb{C}^{\mathcal{P}_{\text{num}}}$ with $\mathcal{P}_{\text{num}} = \mathcal{P}_{\text{hor}}\mathcal{P}_{\text{ver}}$. However later on it will be more sensible to consider this as a vector. In order to guarantee a unified access we will use the x as member of the actual base space $\mathbb{B}$, i.e. $\mathcal{P}(x)$ is always well defined.

In standard applications one normally finds $\mathcal{P}_{\text{hor}} = \mathcal{P}_{\text{ver}} = 256$ i.e. we have to reconstruct 65536 variables simultaniously.

2.4 Receiver Sensitivities

We assume to have several receivers $\mathcal{R}$ numbered from 1 to $\mathcal{R}_{\text{num}}$. Due to physical properties of the receivers $\mathcal{R}$, their spatial configuration and other properties we do not get an exact image but a multiplied one with the sensitivity kernel

$$\mathcal{S}_{\mathcal{R}} : \mathbb{B} \to \mathbb{C}$$

2.5 Measurements

We can just measure in the k-space (Fourier transformed space of the image). There we do not want to measure everything but a selected number of lines. I.e we have a subset $\mathbb{M} \subset \mathbb{B}$. The projection operator to this subspace shall be denoted by

$$\mathbf{P} : \mathbb{B} \to \mathbb{M}$$

The number of elements in $\mathbb{M}$ is denoted by $\mathcal{P}_{\text{proj}}$. The measurement is defined as a receiver dependant function

$$\mathcal{M}_{\mathcal{R}} : \mathbb{M} \to \mathbb{C}$$

While measuring we have the following relation:

$$\mathcal{M}_{\mathcal{R}} = \mathbf{P}\,\text{FFT}\,(\mathcal{P} \times \mathcal{S}_{\mathcal{R}})$$

2.6 Sensitivity Kernel

We assume that the sensitivity kernels can be described by a small number of basis functions, i.e. there is a set $\{\mathcal{B}_n\}_{n\in\{1,\dots,\mathcal{B}_{\text{num}}\}}$ such that there are coefficients $b_{\mathcal{R},n}$ with

$$\mathcal{S}_{\mathcal{R}} = \sum_{n=1}^{\mathcal{B}_{\text{num}}} b_{\mathcal{R},n}\mathcal{B}_n$$

Without loss of generality we assume that we can use the same set of basis functions for each of the different receivers. I.e. for each measurement we have using the linearity of the operators

$$\mathcal{M}_{\mathcal{R}} = \sum_{n=1}^{\mathcal{B}_{\text{num}}} b_{\mathcal{R},n}\mathbf{P}\left(\text{FFT}\left(\mathcal{P}\times\mathcal{B}_n\right)\right)$$

The vector of coefficients for the receiver $\mathcal{R}$ will be denoted by $b_{\mathcal{R}}$

3 Calculations

Later on we will see that we need three quantities simultaneously for which we need fast evaluation procedures for the forward operator, the Fréchet derivative and the Fréchet derivative times its adjoint

As a short notation of $\begin{pmatrix}\mathcal{P} & b_1 & \dots & b_{\mathcal{R}_{\text{num}}}\end{pmatrix}$ we will use $\mathcal{X}$.

3.1 Forward Operator

This is actually already the low-cost forward operator computation with $O(\ln(\mathcal{P}_{\text{num}}) * \mathcal{P}_{\text{num}} * \mathcal{R}_{\text{num}})$:

$$\mathbf{F}\begin{pmatrix}\mathcal{P}\\ b_1\\ \vdots\\ b_{\mathcal{R}_{\text{num}}}\end{pmatrix} = \begin{pmatrix}\sum_{n=1}^{\mathcal{B}_{\text{num}}} b_{1,n} & \mathbf{P}\left(\text{FFT}\left(\mathcal{P}\times\mathcal{B}_n\right)\right)\\ & \vdots\\ \sum_{n=1}^{\mathcal{B}_{\text{num}}} b_{\mathcal{R}_{\text{num}},n}\mathbf{P}\left(\text{FFT}\left(\mathcal{P}\times\mathcal{B}_n\right)\right)\end{pmatrix}$$

or shorter if we denote

$$\text{FFT}(\mathcal{P},\mathcal{B}) = \begin{pmatrix}\text{FFT}\left(\mathcal{P}\times\mathcal{B}_1\right) & \cdots & \text{FFT}\left(\mathcal{P}\times\mathcal{B}_{\mathcal{B}_{\text{num}}}\right)\end{pmatrix}$$

then

$$\mathbf{F}\begin{pmatrix}\mathcal{P}\\ b_1\\ \vdots\\ b_{\mathcal{R}_{\text{num}}}\end{pmatrix} = \begin{pmatrix}\mathbf{P}\,\text{FFT}(\mathcal{P},\mathcal{B})b_1\\ \vdots\\ \mathbf{P}\,\text{FFT}(\mathcal{P},\mathcal{B})b_{\mathcal{R}_{\text{num}}}\end{pmatrix} = \begin{pmatrix}\mathbf{P}\,\text{FFT}(\mathcal{P},\mathcal{S}_1)\\ \vdots\\ \mathbf{P}\,\text{FFT}(\mathcal{P},\mathcal{S}_{\mathcal{R}_{\text{num}}})\end{pmatrix}$$

3.2 Fréchet Derivative

For the Fréchet derivative we have to reinterpret FFT as a matrix, i.e.

$$\mathrm{FFT}(\mathcal{P}) = \begin{pmatrix} f_{1,1} & \cdots & f_{1,\mathcal{P}_{\mathrm{num}}} \\ \vdots & \ddots & \vdots \\ f_{\mathcal{P}_{\mathrm{num}},1} & \cdots & f_{\mathcal{P}_{\mathrm{num}},\mathcal{P}_{\mathrm{num}}} \end{pmatrix} \begin{pmatrix} \mathcal{P}_1 \\ \vdots \\ \mathcal{P}_{\mathcal{P}_{\mathrm{num}}} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{\mathcal{P}_{\mathrm{num}}} f_{i,n}\mathcal{P}_i \\ \vdots \\ \sum_{i=1}^{\mathcal{P}_{\mathrm{num}}} f_{\mathcal{P}_{\mathrm{num}},i}\mathcal{P}_i \end{pmatrix}$$

Restricting our attention to the receiver $\mathcal{R}$ first and denoting the entries of $\mathcal{S}_{\mathcal{R}}$ by $\mathcal{S}_{\mathcal{R},i}$, we have:

$$\mathbf{F}_{\mathcal{R}}(\mathcal{X}) = \mathbf{P}\left(\mathrm{FFT}\left(\mathcal{P} \times \mathcal{S}_{\mathcal{R}}\right)\right) = \mathbf{P} \begin{pmatrix} \sum_{i=1}^{\mathcal{P}_{\mathrm{num}}} f_{1,i} \quad \mathcal{S}_{\mathcal{R},i}\mathcal{P}_i \\ \vdots \\ \sum_{i=1}^{\mathcal{P}_{\mathrm{num}}} f_{\mathcal{P}_{\mathrm{num}},i}\mathcal{S}_{\mathcal{R},i}\mathcal{P}_i \end{pmatrix}$$

Hence we have

$$\frac{\partial}{\partial b_{\mathcal{R},n}}\mathbf{F}[\mathcal{X}] = \mathbf{P}\left(\mathrm{FFT}\left(\mathcal{P} \times \mathcal{B}_n\right)\right)$$

and

$$\frac{\partial}{\partial \mathcal{P}_k}\mathbf{F}_{\mathcal{R}}[\mathcal{X}] = \mathbf{P} \begin{pmatrix} f_{1,k} \quad \mathcal{S}_{\mathcal{R},k} \\ \vdots \\ f_{\mathcal{P}_{\mathrm{num}},i}\mathcal{S}_{\mathcal{R},k} \end{pmatrix} = \mathbf{P}\left(\mathrm{FFT}(1_k)\right)\mathcal{S}_{\mathcal{R},k}$$

where 1_k denotes the image with just position k is 1 and all others 0. Hence we have

$$\mathrm{FFT}(1_k) = \begin{pmatrix} f_{1,k} \\ \vdots \\ f_{\mathcal{P}_{\mathrm{num}},k} \end{pmatrix}$$

Now, if the whole collection of $\mathrm{FFT}(1_k)$ is called $\mathrm{FFT}(\mathbf{1})$ we get

$$\mathbf{F}_{\mathcal{R}}[\mathcal{X}]' = \big(\mathbf{P}\left(\mathrm{FFT}(\mathbf{1})\right) \times \mathcal{S}_{\mathcal{R}} \quad \mathbf{P}\left(\mathrm{FFT}\left(\mathcal{P} \times \mathcal{B}_1\right)\right) \quad \cdots \quad \mathbf{P}\left(\mathrm{FFT}\left(\mathcal{P} \times \mathcal{B}_{\mathcal{B}_{\mathrm{num}}}\right)\right)\big)$$

or shorter

$$\mathbf{F}_{\mathcal{R}}[\mathcal{X}]' = \big(\mathbf{P}\left(\mathrm{FFT}(\mathbf{1})\right) \times \mathcal{S}_{\mathcal{R}} \quad \mathbf{P}\,\mathrm{FFT}(\mathcal{P},\mathcal{B})\big)$$

So finally we get

$$\mathbf{F}[\mathcal{X}]' = \begin{pmatrix} \mathbf{P}\left(\mathrm{FFT}(\mathbf{1})\right) \times \mathcal{S}_1 & \ldots & \mathbf{P}\,\mathrm{FFT}(\mathcal{P},\mathcal{B}) & \ldots & 0 & \ldots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{P}\left(\mathrm{FFT}(\mathbf{1})\right) \times \mathcal{S}_{\mathcal{R}} & \ldots & 0 & \ldots & \mathbf{P}\,\mathrm{FFT}(\mathcal{P},\mathcal{B}) & \ldots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{P}\left(\mathrm{FFT}(\mathbf{1})\right) \times \mathcal{S}_{\mathcal{R}_{\mathrm{num}}} & \ldots & 0 & \ldots & 0 & \ldots & \mathbf{P}\,\mathrm{FFT}(\mathcal{P},\mathcal{B}) \end{pmatrix}$$

In order to have a shorter notation for the next section we will denote the submatrices/tensors $\mathbf{D_1} = \mathbf{P}\,(\mathrm{FFT}(\mathbf{1}))$ and $\mathbf{D}_\mathcal{P} = \mathbf{P}\,\mathrm{FFT}(\mathcal{P}, \mathcal{B})$. This yields

$$\mathbf{F}[\mathcal{X}]' = \begin{pmatrix} \mathbf{D_1} \times \mathcal{S}_1 & \dots & \mathbf{D}_\mathcal{P} & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{D_1} \times \mathcal{S}_\mathcal{R} & \dots & 0 & \dots & \mathbf{D}_\mathcal{P} & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{D_1} \times \mathcal{S}_{\mathcal{R}_\mathrm{num}} & \dots & 0 & \dots & 0 & \dots & \mathbf{D}_\mathcal{P} \end{pmatrix}$$

3.3 Fréchet Derivative times its adjoint

We want to compute $\left(\mathbf{F}[\mathcal{X}]'\right)^T \mathbf{F}[\mathcal{X}]'$ in an efficient way.

$$\begin{pmatrix} \mathcal{S}_1^T \times \mathbf{D_1}^T & \dots & \mathcal{S}_\mathcal{R}^T \times \mathbf{D_1}^T & \dots & \mathcal{S}_{\mathcal{R}_\mathrm{num}}^T \times \mathbf{D_1}^T \\ \vdots & & \vdots & & \vdots \\ \mathbf{D}_\mathcal{P}^T & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & \mathbf{D}_\mathcal{P}^T & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & \mathbf{D}_\mathcal{P}^T \end{pmatrix} \begin{pmatrix} \mathbf{D_1} \times \mathcal{S}_1 & \dots & \mathbf{D}_\mathcal{P} & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{D_1} \times \mathcal{S}_\mathcal{R} & \dots & 0 & \dots & \mathbf{D}_\mathcal{P} & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{D_1} \times \mathcal{S}_{\mathcal{R}_\mathrm{num}} & \dots & 0 & \dots & 0 & \dots & \mathbf{D}_\mathcal{P} \end{pmatrix}$$

$$= \begin{pmatrix} \left(\sum_{\mathcal{R}=1}^{\mathcal{R}_\mathrm{num}} \mathcal{S}_\mathcal{R} \mathcal{S}_\mathcal{R}^T\right) \times \mathbf{D_1}^T\mathbf{D_1} & \mathcal{S}_1^T \times \mathbf{D_1}^T\mathbf{D}_\mathcal{P} & \dots & \mathcal{S}_\mathcal{R}^T \times \mathbf{D_1}^T\mathbf{D}_\mathcal{P} & \dots & \mathcal{S}_{\mathcal{R}_\mathrm{num}}^T \times \mathbf{D_1}^T\mathbf{D}_\mathcal{P} \\ \mathbf{D}_\mathcal{P}^T\mathbf{D_1} \times \mathcal{S}_1 & \mathbf{D}_\mathcal{P}^T\mathbf{D}_\mathcal{P} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{D}_\mathcal{P}^T\mathbf{D_1} \times \mathcal{S}_\mathcal{R} & 0 & \dots & \mathbf{D}_\mathcal{P}^T\mathbf{D}_\mathcal{P} & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{D}_\mathcal{P}^T\mathbf{D_1} \times \mathcal{S}_{\mathcal{R}_\mathrm{num}} & 0 & \dots & 0 & \dots & \mathbf{D}_\mathcal{P}^T\mathbf{D}_\mathcal{P} \end{pmatrix}$$

3.4 Analysis

Hence one needs to do the following computations. First we note the (rough) number of operations for the ones which need to be performed once:

- $\mathbf{D_1} = \mathbf{P}\,(\mathrm{FFT}(\mathbf{1}))$: $\log(\mathcal{P}_\mathrm{num})\mathcal{P}_\mathrm{num}{}^2$
- $\mathbf{D_1}^T\mathbf{D_1}$: $\mathcal{P}_\mathrm{num}{}^2\mathcal{P}_\mathrm{proj}$

I.e. together approximately $\mathcal{P}_\mathrm{proj}\mathcal{P}_\mathrm{num}{}^2$. Now the operations which need to be done everytime:

- $\mathbf{P}\,\mathrm{FFT}(\mathcal{P}, \mathcal{B})$: $\mathcal{B}_\mathrm{num}\mathcal{P}_\mathrm{num}\log(\mathcal{P}_\mathrm{num})$
- all $\mathbf{P}\,\mathrm{FFT}(\mathcal{P}, \mathcal{S}_\mathcal{R})$: $\mathcal{R}_\mathrm{num}\mathcal{P}_\mathrm{num}\log(\mathcal{P}_\mathrm{num})$
- all $\mathbf{P}\,(\mathrm{FFT}(\mathbf{1})) \times \mathcal{S}_\mathcal{R}$: $\mathcal{P}_\mathrm{num}\mathcal{P}_\mathrm{proj}\mathcal{R}_\mathrm{num}$
- $\left(\sum_{\mathcal{R}=1}^{\mathcal{R}_\mathrm{num}} \mathcal{S}_\mathcal{R}\mathcal{S}_\mathcal{R}^T\right) \times \mathbf{D_1}^T\mathbf{D_1}$: $\mathcal{P}_\mathrm{num}{}^2\mathcal{R}_\mathrm{num}$

- $\mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathcal{P}}$: $\mathcal{B}_{\text{num}} \mathcal{P}_{\text{proj}} \mathcal{P}_{\text{num}}$
- all $\mathcal{S}_{\mathcal{R}}^T \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathcal{P}}$: $\mathcal{R}_{\text{num}} \mathcal{P}_{\text{num}} \mathcal{B}_{\text{num}}$
- $\mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}}$: ${\mathcal{B}_{\text{num}}}^2 \mathcal{P}_{\text{proj}}$

I.e. together approximately $\mathcal{R}_{\text{num}} {\mathcal{P}_{\text{num}}}^2$.

Furthermore we need to invert the matrix $\left(\mathbf{F}[\mathcal{X}]'\right)^T \mathbf{F}[\mathcal{X}]'$ later on (respectively a matrix of the same size): Using Cholesky factorization we need $3\left(\mathcal{P}_{\text{num}} + \mathcal{B}_{\text{num}} \mathcal{R}_{\text{num}}\right)^3$ operations and is so the by far most cost intensive operation.

Please note that we can perhaps save some operations by rewriting the algorithms. E.g. as we will not have to compute $\left(\mathbf{F}[\mathcal{X}]'\right)^T \mathcal{Y}$ often we do not really have to compute the whole matrix $\left(\mathbf{F}[\mathcal{X}]'\right)^T$.

3.5 Remarks

Assume that one measures all measurements with double intensiveness. This can have two reasons. Either all receivers measure with double intensiveness or the picture is double as bright. We have no possibility whatsoever to distinguish between these two cases. Therefore we need at least one additional equation.

We want to have a rather stable additional quantity which does not significantly reduce the speed of our computations. Therefore we propose to use the sum of each sensitivity distribution $\mathcal{S}_{\mathcal{R}}$. This has two advantages. On the one hand its the only quantity which has an derivative independent of the actual values of $\mathcal{S}_{\mathcal{R}}$ and on the other hand small errors should average themselves due to the law of large numbers.

3.6 Fast Forward Computation of the Fréchet Derivative

If we like to do a fast forward computation of the Fréchet derivative we can exploit the following structure

$$\begin{pmatrix} \mathbf{D}_{\mathbf{1}} \times \mathcal{S}_1 & \dots & \mathbf{D}_{\mathcal{P}} & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}} & \dots & 0 & \dots & \mathbf{D}_{\mathcal{P}} & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}_{\text{num}}} & \dots & 0 & \dots & 0 & \dots & \mathbf{D}_{\mathcal{P}} \end{pmatrix} \begin{pmatrix} \overline{\mathcal{P}} \\ \overline{b_1} \\ \vdots \\ \overline{b_{\mathcal{R}_{\text{num}}}} \end{pmatrix} = \begin{pmatrix} \left(\mathbf{D}_{\mathbf{1}} \times \mathcal{S}_1\right) \overline{\mathcal{P}} + \mathbf{D}_{\mathcal{P}} \overline{b_1} \\ \vdots \\ \left(\mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}}\right) \overline{\mathcal{P}} + \mathbf{D}_{\mathcal{P}} \overline{b_{\mathcal{R}}} \\ \vdots \\ \left(\mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}_{\text{num}}}\right) \overline{\mathcal{P}} + \mathbf{D}_{\mathcal{P}} \overline{b_{\mathcal{R}_{\text{num}}}} \end{pmatrix}$$

Using the relation

$$\left(\mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}}\right) \overline{\mathcal{P}} = \left(\mathbf{D}_{\mathbf{1}} \times \overline{\mathcal{P}}\right) \mathcal{S}_{\mathcal{R}}$$

we need $\mathcal{R}_{\text{num}} \mathcal{P}_{\text{num}} \mathcal{P}_{\text{proj}} + \mathcal{R}_{\text{num}} \mathcal{B}_{\text{num}} \mathcal{P}_{\text{proj}}$

3.7 Fast Forward Computation of the Adjoint of the Fréchet Derivative

If we want to have a fast forward computation of the adjoint of the Fréchet Derivative we do not necessarily need to compute the matrix first. Obviously the solution has the following structure

$$\begin{pmatrix} \mathcal{S}_1^T \times \mathbf{D}_{\mathbf{1}}^T & \dots & \mathcal{S}_{\mathcal{R}}^T \times \mathbf{D}_{\mathbf{1}}^T & \dots & \mathcal{S}_{\mathcal{R}_{\mathrm{num}}}^T \times \mathbf{D}_{\mathbf{1}}^T \\ \vdots & & \vdots & & \vdots \\ \mathbf{D}_{\mathcal{P}}^T & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & \mathbf{D}_{\mathcal{P}}^T & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & \mathbf{D}_{\mathcal{P}}^T \end{pmatrix} \begin{pmatrix} \mathcal{M}_1 \\ \vdots \\ \mathcal{M}_{\mathcal{R}} \\ \vdots \\ \mathcal{M}_{\mathcal{R}_{\mathrm{num}}} \end{pmatrix} = \begin{pmatrix} \sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} (\mathcal{S}_{\mathcal{R}}^T \times \mathbf{D}_{\mathbf{1}}^T) \mathcal{M}_{\mathcal{R}} \\ \mathbf{D}_{\mathcal{P}}^T \mathcal{M}_1 \\ \vdots \\ \mathbf{D}_{\mathcal{P}}^T \mathcal{M}_{\mathcal{R}} \\ \vdots \\ \mathbf{D}_{\mathcal{P}}^T \mathcal{M}_{\mathcal{R}_{\mathrm{num}}} \end{pmatrix}$$

Using

$$\sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} (\mathcal{S}_{\mathcal{R}}^T \times \mathbf{D}_{\mathbf{1}}^T) \mathcal{M}_{\mathcal{R}} = \left(\left(\sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} \mathcal{S}_{\mathcal{R}}^T \mathcal{M}_{\mathcal{R}} \right) \times \mathbf{D}_{\mathbf{1}}^T \right) \mathbf{1}$$

we have an algorithm just demanding $\mathcal{R}_{\mathrm{num}} \mathcal{P}_{\mathrm{num}} \mathcal{P}_{\mathrm{proj}} + \mathcal{R}_{\mathrm{num}} \mathcal{P}_{\mathrm{proj}} \mathcal{B}_{\mathrm{num}}$ steps.

3.8 Fast computation of the Forward Solution of Fréchet Derivative times its adjoint

Now we will see how much time one looses by directly computing both of the above together

$$\begin{pmatrix} \left(\sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} \mathcal{S}_{\mathcal{R}} \mathcal{S}_{\mathcal{R}}^T \right) \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathbf{1}} & \mathcal{S}_1^T \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathcal{P}} & \dots & \mathcal{S}_{\mathcal{R}}^T \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathcal{P}} & \dots & \mathcal{S}_{\mathcal{R}_{\mathrm{num}}}^T \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathcal{P}} \\ \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathbf{1}} \times \mathcal{S}_1 & \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}} & 0 & \dots & \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}} & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathbf{1}} \times \mathcal{S}_{\mathcal{R}_{\mathrm{num}}} & 0 & \dots & 0 & \dots & \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}} \end{pmatrix}$$

$$\begin{pmatrix} \overline{\mathcal{P}} \\ \overline{b_1} \\ \vdots \\ \overline{b_{\mathcal{R}_{\mathrm{num}}}} \end{pmatrix} = \begin{pmatrix} \left(\left(\sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} \mathcal{S}_{\mathcal{R}} \mathcal{S}_{\mathcal{R}}^T \right) \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathbf{1}} \right) \overline{\mathcal{P}} + \left(\left(\sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} \mathcal{S}_{\mathcal{R}}^T \overline{b_{\mathcal{R}}} \right) \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathcal{P}} \right) \mathbf{1} \\ \left(\mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathbf{1}} \times \overline{\mathcal{P}} \right) \mathcal{S}_1 + \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}} \overline{b_1} \\ \vdots \\ \left(\mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathbf{1}} \times \overline{\mathcal{P}} \right) \mathcal{S}_{\mathcal{R}} + \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}} \overline{b_{\mathcal{R}}} \\ \vdots \\ \left(\mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathbf{1}} \times \overline{\mathcal{P}} \right) \mathcal{S}_{\mathcal{R}_{\mathrm{num}}} + \mathbf{D}_{\mathcal{P}}^T \mathbf{D}_{\mathcal{P}} \overline{b_{\mathcal{R}_{\mathrm{num}}}} \end{pmatrix}$$

Just computing $\left(\left(\sum_{\mathcal{R}=1}^{\mathcal{R}_{\mathrm{num}}} \mathcal{S}_{\mathcal{R}} \mathcal{S}_{\mathcal{R}}^T \right) \times \mathbf{D}_{\mathbf{1}}^T \mathbf{D}_{\mathbf{1}} \right) \overline{\mathcal{P}}$ already takes $\mathcal{R}_{\mathrm{num}} {\mathcal{P}_{\mathrm{num}}}^2$ operations. I.e. it is by far more cost intensive than computing the two parts separately. Actually, there is almost no difference between computing this and matrix times vector, so if we have access to the matrix we better use this one.

4 Reconstruction Algorithms

Our main task is the reconstruction of the image $\mathcal{P}$. However, as we have seen we can just measure the image together with the unknown receiver sensitivities. Therefore it is impossible to gain the knowledge of $\mathcal{P}$ without immediately also getting the coefficients $b_{\mathcal{R},n}$ and vice versa.

4.1 Various Algorithms and Problem Interpretations

There are several one step algorithms, which provide solutions to the problem. Among them are GRAPPA and SENSE. However we will later stick to the interpretation as a non-linear inverse problem.

4.1.1 SENSE

SENSitivity Encoding assumes perfect knowledge about the spatial sensitivity profile of the array detector elements. The original paper [2] gives some instructions on how to estimate these spatial sensitivity profiles, but it is in general a non-trivial problem in itself. Given perfect knowledge, however, the PPI reconstruction problem reduces to a linear system of equations with a noise disturbance. In its simplest form, the SENSE experiment yields a regularly undersampled data set where the number of k-space lines in the full encoding is an integer multiple of the acceleration factor. After Fourier transformation of the undersampled data set, each pixel in the resulting image is the sensitivity weighted superposition of the unknown true image pixels. The full linear system of equations reduces to one small linear system of equations per pixel in the intermediate image.

4.1.2 GRAPPA

GeneRAlized Autocalibrating Partially Parallel Acquisitions assumes that the missing k-space samples of one array detector data set can be expressed as a linear combination of neighboring k-space samples in all array detector data sets. This assumption is motivated by the fact that the modulation of the unknown true image intensity distribution with the spatial sensitivity profiles, which is a convolution in k-space between the spatial frequency spectra of the true image and the sensitivity profiles, will locally distribute the information of any k-space sample among its neighbors. Sensitivity profiles can be generally assumed to be smooth functions in image space, and hence the corresponding convolution kernels have compact support in k-space.

For the typically very regular undersampling schemes, the coefficients of the linear combination reconstruction, i.e., the reconstruction convolution kernels, can be estimated from a sufficiently large fully sampled portion of the k-space data acquired with the same detector array. It has interestingly been shown that the calibration does not have to be performed with the same imaging parameters as the undersampled experiment, but can be of quite a different image contrast. However, in most cases it is beneficial to make the calibration acquisition part of the imaging experiment itself, i.e., autocalibration. Typically, a few extra k-space lines in the center of k-space are acquired.

GRAPPA suffers from the large range of possible choices for the reconstruction kernel parameters, like number of support points in the individual k-space directions, and an infinite variety of possibilities how to use the available calibration data for kernel coefficient estimation. This dilemma has been treated in [7]. It has on the other hand the capability of high-quality image reconstruction, even when the spatial resolution of the calibration measurement is inadequate for the spatial frequency content of the sensitivity profiles, see [8].

4.1.3 Non-linear Inverse Problem

From a mathematical point of view the most sensible thing would be generating the receiver sensitivities and the image $\mathcal{P}$ in one go. The big disadvantage is that we need to handle this as a non-linear inverse problem. However this will enable us to also consider the influence of noise.

Therefore we will need the following notation. The solution is denoted, as beforehand, by

$$\mathcal{X} = (\mathcal{P}, \mathcal{S}_1, \ldots, \mathcal{S}_{\mathcal{R}_{\text{num}}})$$

and the exact data will be denoted by $\mathcal{Y}$, the measured ones by $\mathcal{Y}^\delta$.

4.2 Non-linear Inverse Problems

Many inverse problems, not just the one formulated beforehand, can be formulated as nonlinear operator equations

$$\mathbf{F}(\mathcal{X}) = \mathcal{Y}$$

with an injective operator $\mathbf{F} : D(\mathbf{F}) \subset \mathbb{X} \to \mathbb{Y}$ between the Hilbert spaces $\mathbb{X}, \mathbb{Y}$. We assume that $\mathbf{F}$ is Fréchet differentiable on its domain $D(\mathbf{F})$ and that the measured data $\mathcal{Y}^\delta$ are perturbed by noise with noise level δ, i.e.

$$||\mathcal{Y} - \mathcal{Y}^\delta|| \leq \delta.$$

These problems are ill-posed in the sense that the solution does not depend continuously on the data, i.e. $\mathbf{F}^{-1}$ is not continuous.

There are a number of different solution algorithms for non-linear inverse problems [5]. The most prominent ones which require a matrix inversion are the iteratively regularized Gauß-Newton method and the very similar Levenberg-Marquardt method.

The Landweber method is comparably slower but does not require the inversion of a matrix. For linear problems there also exist faster methods, e.g. ν-methods which have no convergence proofs for non-linear problems.

Common to all of these methods is that they require an initial guess $\mathcal{X}_0$. The better the initial guess, the better and faster the method gets.

4.3 IRGN-Method

One of the most successful methods to solve such problem is the *iteratively regularized Gauß-Newton method* (IRGNM) suggested by Bakushinskii [9]. The n-th step of this methods consists in applying Tikhonov regularization to the linearized operator equation

$$\mathbf{F}'[\mathcal{X}_n^\delta](\mathcal{X}_{n+1}^\delta - \mathcal{X}_n^\delta) = \mathcal{Y}^\delta - \mathbf{F}(\mathcal{X}_n^\delta)$$

with the initial guess $\mathcal{X}_0$:

$$\mathcal{X}_{n+1}^{\delta} = \underset{x \in \mathcal{X}}{\operatorname{armin}} \|\mathbf{F}'[\mathcal{X}_n^{\delta}](\mathcal{X} - \mathcal{X}_n^{\delta}) - \mathcal{Y}^{\delta} + \mathbf{F}(\mathcal{X}_n^{\delta})\|^2 + \alpha_n \|\mathcal{X} - \mathcal{X}_0\|^2$$

For simplicity we only consider the following a-priori choice of the regularization parameters α_n:

$$\alpha_n = \alpha_0 q^n, \qquad q \in (0,1), \alpha_0 > 0$$

The regularity of the solution relative to the smoothing properties of the operator is measured in terms of source conditions of the form

$$\mathcal{X}_0 - \mathcal{X} = \Lambda(\mathbf{F}'[\mathcal{X}]^* \mathbf{F}'[\mathcal{X}])\mathcal{W}, \qquad \|\mathcal{W}\| \leq \varrho$$

where $\Lambda : [0, \|\mathbf{F}'[\mathcal{X}]\|^2] \to \mathbb{R}$ is a monotonic increasing function satisfying $\Lambda(0) = 0$. It is always assumed that the scaling condition

$$\|\mathbf{F}'[\mathcal{X}]\| \leq C_{\mathrm{s}} < 1$$

is satisfied, which can always be achieved by rescaling the norm in $\mathbb{Y}$.

The IRGNM for converges with optimal rate (subject to the qualification of Tikhonov regularization itself) under rather general conditions which has been shown in [6] and the references therein. Kaltenbacher [10] suggested to replace Tikhonov regularization by a regularization method with higher qualification and consider iteration schemes of the form

$$\mathcal{X}_{n+1}^{\delta} := \mathcal{X}_0 + g_{\alpha_n}\left(\mathbf{F}'[\mathcal{X}_n^{\delta}]^* \mathbf{F}'[\mathcal{X}_n^{\delta}]\right) \mathbf{F}'[\mathcal{X}_n^{\delta}]^* \left(\mathcal{Y}^{\delta} - \mathbf{F}(\mathcal{X}_n^{\delta}) + \mathbf{F}'[\mathcal{X}_n^{\delta}](\mathcal{X}_n^{\delta} - \mathcal{X}_0)\right).$$

In particular, for K-times iterated Tikhonov regularization we have $g_{\alpha}(\lambda) = \frac{(\lambda+\alpha)^K - \alpha^K}{\lambda(\lambda+\alpha)^K}$.

4.3.1 Parameter Choice

A crucial ingredient of any iterative regularization method is an appropriate stopping rule. However, the best image in a norm sense does not need to be an optimal image for medical applications; actually normally it is not. As in many real situations the noise and the other problem parameters do not change considerably from measurement to measurement it might be more sensible to determine a good regularization parameter experimentally.

4.3.2 Matrix inversion

As we see this method requires matrix inversion methods. In principle there are two big different possibilities. The one is using standard full inversion algorithms, the best to use would be Cholesky inversion because of the properties of the matrix [11].

```
$\mathbf{G} \leftarrow \mathrm{Chol}\left(\mathbf{F}'[\mathcal{X}_n^\delta]^*\mathbf{F}'[\mathcal{X}_n^\delta] + \alpha_n\mathbf{Id}\right)$
$\mathcal{X}_{hilf} \leftarrow \mathbf{F}'[\mathcal{X}_n^\delta]^* \left(\mathcal{Y}^\delta - \mathbf{F}(\mathcal{X}_n^\delta)\right)$
$\mathcal{X}_{update} \leftarrow \mathcal{X}_0 - \mathcal{X}_n$
for $k \leftarrow 1 \dots K$
    $\mathcal{X}_{input} \leftarrow \mathcal{X}_{hilf} + \alpha_n\mathcal{X}_{update}$
    $\mathcal{X}_{update} \leftarrow \mathbf{G}^{-1}\left(\mathbf{G}^*\right)^{-1}\mathcal{X}_{input}$
end
$\mathcal{X}_{n+1} \leftarrow \mathcal{X}_n + \mathcal{X}_{update}$
```

However as we do perturbed inversions anyway it is not necessary that the inversion of the matrix itself is completely exact. Therefore one can alternatively use the CG (Conjugate Gradient)-method in order to invert the matrix [11]. ε is a tuning parameter.

```
$\mathbf{G} \leftarrow \mathbf{F}'[\mathcal{X}_n^\delta]^*\mathbf{F}'[\mathcal{X}_n^\delta] + \alpha_n\mathbf{Id}$
$\mathcal{X}_{update} \leftarrow \mathbf{F}'[\mathcal{X}_n^\delta]^* \left(\mathcal{Y}^\delta - \mathbf{F}(\mathcal{X}_n^\delta)\right) + \alpha_n\left(\mathcal{X}_0 - \mathcal{X}_n\right)$
$\mathcal{X}_{2nd} \leftarrow 0$
$norm_{old} \leftarrow \infty$
while $\|\mathcal{X}_{update}\| > \alpha_n\varepsilon\|\mathcal{X}_{2nd}\|$
    $\mathcal{X}_{CG} \leftarrow \mathcal{X}_{update} + \frac{\|\mathcal{X}_{update}\|^2}{norm_{old}^2}\mathcal{X}_{CG}$
    $\mathcal{X}_{Hilf} \leftarrow \mathbf{G}\mathcal{X}_{CG}$
    $\mathcal{X}_{2nd} \leftarrow \mathcal{X}_{2nd} + \frac{\|\mathcal{X}_{update}\|^2}{\langle\mathcal{X}_{CG}, \mathcal{X}_{Hilf}\rangle}\mathcal{X}_{CG}$
    $norm_{old} \leftarrow \|\mathcal{X}_{update}\|$
    $\mathcal{X}_{update} \leftarrow \mathcal{X}_{update} - \frac{norm_{old}^2}{\langle\mathcal{X}_{CG}, \mathcal{X}_{Hilf}\rangle}\mathcal{X}_{Hilf}$
end
$\mathcal{X}_{n+1} \leftarrow \mathcal{X}_n + \mathcal{X}_{update}$
```

As we see this method does not require the inversion of the regularization matrix, however iterated Tikhonov is not sensible in this case and therefore omitted in the above algorithm.

4.3.3 Speed

The IRGN method exhibits an exponential convergence speed, i.e. if δ gets smaller by a factor of q, the number of necessary iteration steps just augments by one.

4.4 Landweber Method

The Landweber method is defined as follows (ε is again a tuning parameter):

$$\begin{aligned} \mathcal{X}_{hilf} &\leftarrow \mathbf{F}'[\mathcal{X}_n^\delta]^* \left(\mathcal{Y}^\delta - \mathbf{F}(\mathcal{X}_n^\delta)\right) \\ \mathcal{X}_{n+1} &\leftarrow \mathcal{X}_n + \varepsilon \mathcal{X}_{hilf} \end{aligned}$$

respectively the ν method like equation

$$\begin{aligned} \mathcal{X}_{hilf} &\leftarrow \mathbf{F}'[\mathcal{X}_n^\delta]^* \left(\mathcal{Y}^\delta - \mathbf{F}(\mathcal{X}_n^\delta)\right) \\ \mathcal{X}_{n+1} &\leftarrow \mathcal{X}_n + \varepsilon \mathcal{X}_{hilf} + \frac{\varepsilon}{4} \left(\mathcal{X}_n - \mathcal{X}_{n-1}\right) \end{aligned}$$

As we see this method does not require matrix inversions. However this method exhibits just linear speed. In the linear case the ν methods have quadratic speed, to the authors knowledge no results for non-linear problems are known.

4.5 Initial Guess

As we have seen a crucial point is providing an initial guess for all of the proposed methods. Possibilities are manifold and all have their advantages and disadvantages. In particular one has to consider that the solution algorithms just change the starting solution. Hence it is possible that artifacts at least partly persist; on the other hand we have that the nearer the initial guess to the later solution, the smaller the error gets [6]:

- SENSE or GRAPPA
  + fast and well-known
  – it is possible that we end up with the same kind of artifacts we know from SENSE and GRAPPA

- zero solution
  + fast, no artifacts
  – far away from the real solution

- small scale reconstructions
  + very likely to omit any kind of artifacts, near to the final solution
  – slow

5 Numerics

5.1 Sensitivities

A major ingredient for the numerical treatment is the right initial guess for the sensitivities. This decomposes again into two parts. One is estimating this quantity out of real data and the other is finding an appropriate but small basis system $\{\mathcal{B}_n\}_{n\in\{1,\ldots,\mathcal{B}_{\text{num}}\}}$ to represent the sensitivities.

Considering the basis system there are several possibilities. A standard approach is using a small area of the k-space (typically about 0.1%) to model the sensitivities. This works considerably well if one uses a very smooth initial guess.

5.2 Simulation

We intensively tested the method with simulated data. However in terms of reliability we have to keep in mind that the initial “measurement” and the later simulations of measurements needed for the solution procedure were performed in the same way.

We observed that the method worked reliably in this setting, however a good knowledge of the sensitivities was of great advantage. For small noise levels reconstructions could be (in terms of reconstruction error) more than ten times better than GRAPPA which basically tells that the new method seems intrinsically to be much less biased.

Depending on the size and the noise level reconstruction times were considerably long. However, this is just the case for very small noise levels, the more noise the faster the method.

In all our applications we had the impression that IRGN with CG yielded much better results in a shorter time than Landweber iteration. However, due to the speed of one step of the Landweber iteration and the big improvements in the first steps it might still be worthwhile to consider this method to go over the data shortly if time is an issue.

5.3 Application

For the application we used real data acquired on a Siemens MAGNETOM Trio a Tim System clinical whole-body scanner operating at a magnetic flux density of 3.0 Tesla. A full three-dimensional gradient echo data set of the head of a healthy volunteer was acquired using the system’s 12-channel head matrix coil. The field of view was 240mm in all three spatial directions with the frequency encoding in head-feet direction. The resolution was 256 samples in all three spatial directions.

In order to do fast prototyping we used Matlab® 7 where we reconstructed an image with resolution 256×256. We restricted our attention to one slice with these dimensions. For the reconstruction we used 32 of the 256 lines in the middle and an acceleration factor of 3, i.e. additionally every third line.

As comparison method and initial guess we used GRAPPA. In figure 1 we see GRAPPA, the new method and a reference picture using all possible data beside each other. As usual we applied a manual contrast enhancing step for all of these images.

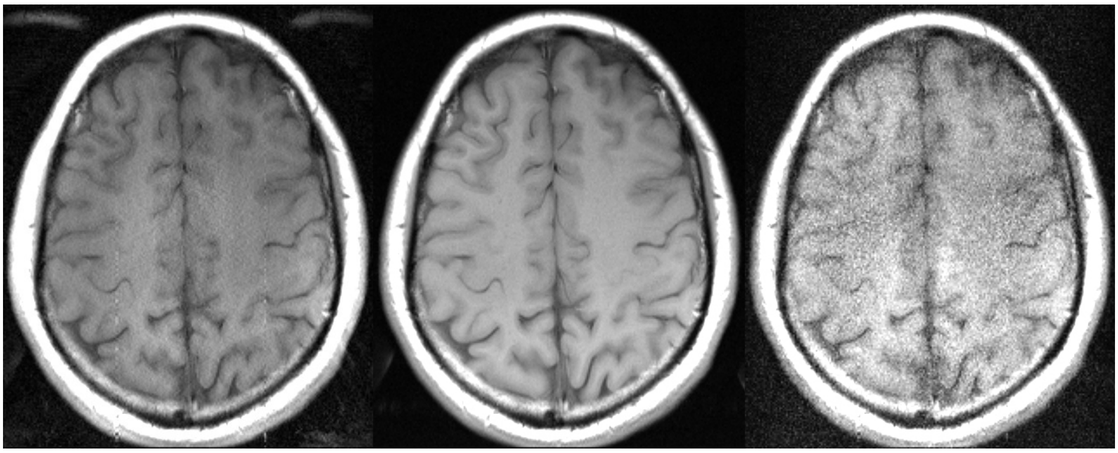

Figure 1: l: New Method m: using all data (reference) r: GRAPPA

5.4 Discussion

From a visual inspection of the results it seems that the new method yields results which are superior to the initial GRAPPA image. An important point is that the GRAPPA reconstruction is medically seen not usable, whereas the new method provides an image which can be used for this purpose. Hence we can alternatively consider this new method as an add-on which can improve images considerably in times where the full processor power of the MRI-machines is not needed.

The proposed method integrates for the first time the sensitivity calibration step and the image reconstruction step in PPI into a single processing step. Therefore, it has a high potential of being superior to all existing methods, especially since the inherent information about the receiver sensitivities which is contained in the undersampled imaging data themselves is also exploited. As an interesting side remark, this method could therefore even work if no additional calibration data are acquired.

Through its general mathematical framework, this method also appears flexible enough to be able to incorporate other a-priori information about the unknown image than sensitivity information of the array detectors. Many methods exploiting structural a-priori information have been proposed in the past for MRI, ranging from real-valuedness over limited support, to model-based approaches with reference images. Being able to incorporate any of these additional a-priori constraints would make the method even more powerful.

Acknowledgements

The support for the first author of the Graduiertenkolleg 1023 of the University of Göttingen is gratefully acknowledged.

References

[1] Sodickson DK, Manning WJ. Simultaneous acquisition of spatial harmonics (SMASH): fast imaging with radiofrequency coil arrays. Magn Reson Med 1997 Oct;38(4):591–603.

[2] Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: sensitivity encoding for fast MRI. Magn Reson Med 1999 Nov;42(5):952–962.

[3] Griswold MA, Jakob PM, Heidemann RM, Nittka M, Jellus V, Wang J, et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA). Magn Reson Med 2002 Jun;47(6):1202–1210.

[4] Kyriakos WE, Panych LP, Kacher DF, Westin CF, Bao SM, Mulkern RV, et al. Sensitivity profiles from an array of coils for encoding and reconstruction in parallel (SPACE RIP). Magn Reson Med 2000 Aug;44(2):301–308.

[5] Engl H, Hanke M, Neubauer A. Regularization of Inverse Problems. Dordrecht, Boston, London: Kluwer Academic Publisher; 1996.

[6] Bauer F, Hohage T. A Lepskij-type stopping rule for regularized Newton methods. Inverse Problems 2005;21:1975–1991.

[7] Huang F, Duensing GR. A theoretical analysis of errors in GRAPPA. Proc Intl Soc Mag Reson Med 2006;14:2468.

[8] Beatty PJ, Brau AC. Understanding the GRAPPA Paradox. Proc Intl Soc Mag Reson Med 2006;14:2467.

[9] Bakushinskii A. The problem of the convergence of the iteratively regularized Gauss-Newton method. Comput Maths Math Phys 1992;32:1353–1359.

[10] Kaltenbacher B. Some Newton-type methods for the regularization of nonlinear ill-posed problems. Inverse Problems 1997;13:729–753.

[11] Golub G, Van Loan C. Matrix computations. 3rd ed. Johns Hopkins Studies in the Mathematical Sciences. Baltimore, MD: Johns Hopkins University Press; 1996.

Institut für Numerische und Angewandte Mathematik
Universität Göttingen
Lotzestr. 16-18
D - 37083 Göttingen

Telefon: 0551/394512
Telefax: 0551/393944

Email: trapp@math.uni-goettingen.de URL: http://www.num.math.uni-goettingen.de

Verzeichnis der erschienenen Preprints:

2006-01	R. Schaback	Limit Problems for Interpolation by Analytic Radial Basis Function
2006-02	N. Bissantz, T. Hohage, A. Munk, F. Ruymgaart	Convergence rates of general regularization methods for statistical inverse problems and applications
2006-03	J. Brimberg, H. Juel, A. Schöbel	Locating a circle on a sphere
2006-04	J. Brimberg, H. Juel, A. Schöbel	Locating a circle on the plane using the minimax criterion
2006-05	L. Ling, R. Opfer, R. Schaback	Results on Meshless Collocation Techniques
2006-06	G. Lube, T. Knopp et.al.	Domain decomposition methods for indoor air flow simulation
2006-07	T. Apel, T. Knopp, G. Lube	Stabilized finite element methods with anisotropic mesh refinement for the Oseen problem
2006-08	R. Schaback	Recovery of Functions From Weak Data Using Unsymmetric Meshless Kernel-Based Methods
2006-09	H.W. Hannacher, S. Horn, A. Schöbel	Stop Location Design in Public Transportation Networks: Covering and Accessibility Objective
2006-10	Q.T. Le Gia, F.J. Narcowich, J.D. Ward, H. Wendland	Continuous and Discrete Least-Squares Approximation by Radial Basis Functions on Spheres
2006-11	R. Schaback	Unsymmetric Meshless Methods for Operator Equations
2006-12	P. Giesl, H. Wendland	Meshless Collocation: Error Estimates with Application to Dynamical Systems
2006-13	H. Wendland	On the stability of meshless symmetric collocation for boundary value problems

2006-14	G. Lube	Stabilized FEM for incompressible flow. Critical review and new trends
2006-15	J. Puerto, A. Schöbel, S. Schwarze	A Class of Infinite Potential Games
2006-16	S. Hein, T. Hohage, W. Koch, J. Schöberl	Acoustic Resonances in High Lift Configuration
2006-17	T. Hohage, M.L. Rapún, F.J. Sayas	Detecting corrosion using thermal measurements
2006-18	H. Wendland	Spatial coupling in aeroelasticity by meshless kernel-based methods
2006-19	F. Bauer, O. Ivanyshyn	Optimal Regularization with Two Interdependent Regularization Parameters
2006-20	F. Bauer	Some Considerations Concerning Regularization and Parameter Choice Algorithms
2006-21	S. Langer, T. Hohage	Convergence analysis of an inexact iteratively regularized Gauss-Newton method under general source conditions
2006-22	H. Haddar, R. Kress	Conformal mapping and an inverse impedance boundary value problem
2006-23	F. Cakoni, R. Kress	Integral equations for inverse problems in corrosion detection from partial Cauchy data
2006-24	H. Eckel, R. Kress	Nonlinear integral equations for the inverse electrical impedance problem
2006-25	F. Bauer, S. Kannengiesser	An alternative approach to the image reconstruction for parallel data acquisition in MRI