



**GyroTools**

[www.gyrotools.com](http://www.gyrotools.com)

# MRsense

**Quickstart**

**SENSE reconstruction with MRecon**

Document Version 2.0, January 2013

## Introduction

The MRsense class is used for calculating the coil sensitivity maps from the SENSE reference scan in an arbitrary geometry. As such the reference scan as well as the (undersampled) SENSE scan is passed to MRsense, which will then reformat the reference scan to the geometry of the SENSE scan. Finally “raw sensitivity maps” are determined by dividing the individual coil images by the bodycoil images. If desired the raw sensitivity maps may be soothed and/or extrapolated to remove any unwanted peaks or singularities which might have created by the division of noisy signals. The necessity to smooth and extrapolate the sensitivities however depends largely on the SENSE unfolding method which is used in the reconstruction afterwards (see next section).

## SENSE implementations in MRecon

Currently there are two different SENSE unfolding methods implemented in MRecon:

- 1) The classical SENSE approach according to: Pruessmann KP, Weiger M, Scheidegger MB, and Boesiger P. SENSE: sensitivity encoding for fast MRI. Magn Reson Med 1999;42:952–962).
- 2) A regularized unfolding method which is very similar to the Scanner implementation.

Mathematically the difference of the two methods lay mostly in the calculation of the unfolded pixel values, which are given by:

Classical:

$$\rho = (S^H \Psi^{-1} S)^{-1} S^H \Psi^{-1} m$$

Regularized:

$$\rho = (S^H \Psi^{-1} S + R^{-1})^{-1} S^H \Psi^{-1} m$$

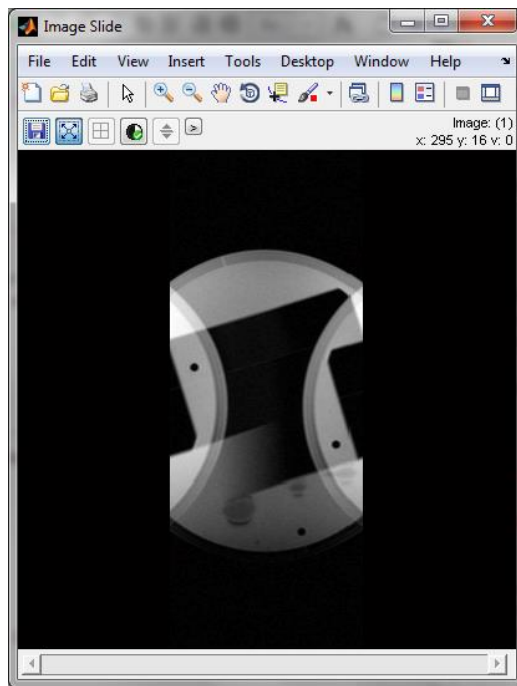
where  $\rho$  contains the unfolded pixel values,  $S$  is the sensitivity matrix,  $\Psi$  the noise covariance matrix, and  $m$  the measured folded pixel values.  $R$  is the regularization matrix which contains the expected values in the unfolded pixels.

The different unfolding procedures have great consequences on the handling and processing of the coil sensitivity maps. While the classical approach needs smooth and noise-free maps to produce adequate results, the regularized approach can take the unprocessed raw sensitivity maps to unfold the SENSE images. Noisy areas and singularities in the sensitivities are punished by the regularization. This behavior makes the regularized approach very robust and easy to use.

## Reconstructing a SENSE scan

Performing a standard MRecon reconstruction on an undersampled scan results in a folded image as illustrated in the following example ( $R = 2$ ):

```
>> r = MRecon('senseR2.lab');
>> r.Perform;
>> r.ShowData;
```



To reconstruct an unfolded image we first have to calculate the coil sensitivity maps and pass it to the reconstruction object.

There are two ways to do that:

- 1) Calculate the sensitivity maps **before** the reconstruction (r.Data is empty)
- 2) Calculate the sensitivity maps **during** the reconstruction process (r.Data is filled)

In the following example we will reconstruct a SENSE scan ( $R=2$ ) using both cases.

## Calculate the Sensitivity maps before the reconstruction

If the sensitivities are calculated before the reconstruction process is started the geometry and matrix size of the maps will be equal to the geometry and matrix size of the undersampled SENSE scan after the Fourier transformation. Meaning that the matrix size will be equal to what is specified in r.Parameter.Encoding.X/Y/ZRes.

However there are two parameters in the MRsense object which will influence the size and “look” of the sensitivity maps: RemoveMOversampling and Rotate (see the following sections for more information). If the first one is set to 1 then the oversampling in readout direction is kept in the sensitivity maps. If the second one is set to 1 then the sensitivity maps are rotated such that the orientation is equal to the final reconstructed REC image which is displayed on the scanner. **These options should be set such that the calculated sensitivity maps match the undersampled SENSE image before the unfolding process.**

In the following it is shown how to calculate the coil sensitivities and unfold the images in MRecon using either the regularizes (default) or classical approach (see previous sections).

## SENSE and scanner software releases

Between scanner release 3 and 4 the acquisition and handling of the SENSE reference scan has changed. Before rel. 4 a single reference scan was acquired containing the individual surface coils as well as the bodycoil, which was sufficient to calculate the coil sensitivities. In release 4 (and higher)

however two reference scans are needed to calculate the coil maps: The SENSE reference scan (containing the surface coils) and the CoilSurvey scan (to calculate a homogeneous reference image). Therefore both prescans have to be exported and passed to MRsense when reconstructing data from release 4 and higher (see below).

### *Regularized approach:*

To calculate the sensitivities before the reconstruction and reconstruct an unfolded image execute the following steps:

- 1) Create a sensitivity object by calling MRsense with the SENSE reference scan and the unfolded scan:

Release 3 and earlier:

```
>> S = MRsense('refscan.lab', 'senseR2.lab');
```

Release 4 and higher:

```
>> S = MRsense('refscan.lab', 'senseR2.lab', coilsurvey.lab');
```

or

Release 3 and earlier:

```
>> ref = MRecon('refscan.lab');  
>> r   = MRecon('senseR2.lab');  
>> S   = MRsense( ref, r );
```

Release 4 and higher:

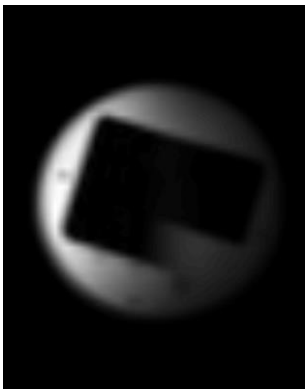
```
>> ref = MRecon('refscan.lab');  
>> coilsurvey = MRecon(coilsurvey.lab);  
>> r     = MRecon('senseR2.lab');  
>> S     = MRsense( ref, r, coilsurvey );
```

- 2) Reformat the reference scan to the geometry of the SENSE scan and calculate the sensitivity maps:

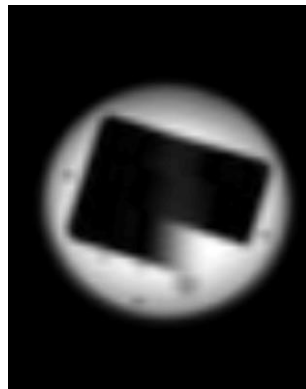
```
>> S.Perform
```

After executing the Perform function, the sensitivity class contains the reformed coil- and bodycoil-images as well as the coil sensitivity maps:

S.ReformattedCoilData



S.ReformattedBodycoilData



S.Sensitivity

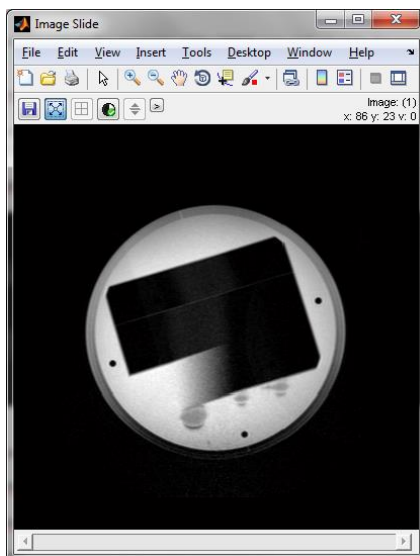


- 3) After calculation pass the MRsense object to the reconstruction object:

```
>> r.Parameter.Recon.Sensitivities = S;
```

- 4) Perform the reconstruction

```
>> r.Perform;
>> r.ShowData;
```



### *Classical approach:*

To calculate the sensitivities before the reconstruction and unfold the image using the classical approach execute the following steps:

- 1) Create a sensitivity object by calling MRsense with the SENSE reference scan and the unfolded scan:

Release 3 and earlier:

```
>> S = MRsense('refscan.lab', 'senseR2.lab');
```

Release 4 and higher:

```
>> S = MRsense('refscan.lab', 'senseR2.lab', coilsurvey.lab');
```

or

Release 3 and earlier:

```
>> ref = MRecon('refscan.lab');
>> r   = MRecon('senseR2.lab');
>> S   = MRsense( ref, r );
```

Release 4 and higher:

```
>> ref = MRecon('refscan.lab');
>> coilsurvey = MRecon(coilsurvey.lab);
>> r     = MRecon('senseR2.lab');
>> S     = MRsense( ref, r, coilsurvey );
```

- 2) Enable the masking, smoothing and extrapolation of the sensitivity maps:

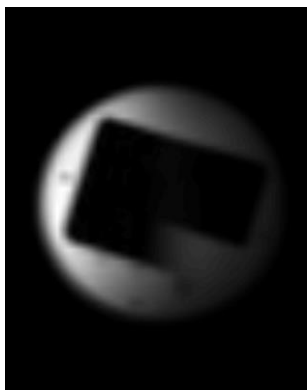
```
>> S.Mask = 1;
>> S.Smooth = 1;
>> S.Extrapolate = 1;
```

- 3) Reformat the reference scan to the geometry of the SENSE scan and calculate the smooth and extrapolated sensitivity maps:

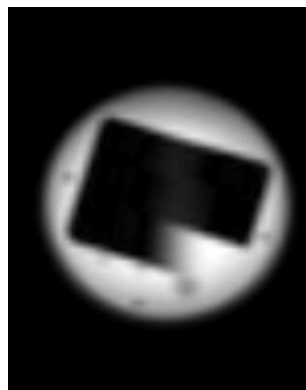
```
>> S.Perform
```

After executing the Perform function, the sensitivity class contains the reformed coil- and bodycoil-images as well as the noise free coil sensitivity maps:

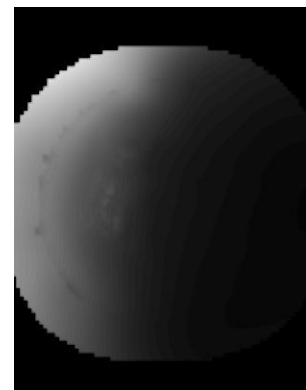
S.ReformattedCoilData



S.ReformattedBodycoilData



S.Sensitivity



- 4) After calculation pass the sensitivity object to the reconstruction object:

```
>> r.Parameter.Recon.Sensitivities = S;
```

- 5) Switch off the regularization:

```
>> r.Parameter.Recon.SENSERegStrength = 0;
```

- 6) Perform the reconstruction

```
>> r.Perform;
```

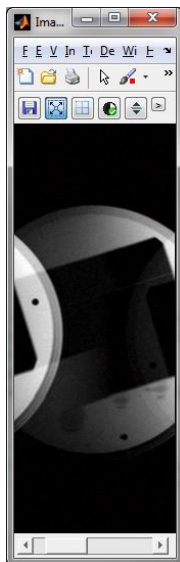
## Calculate the Sensitivity maps during the reconstruction

If the sensitivities are calculated during the reconstruction process the geometry and matrix size of the maps will be equal to the geometry and matrix size of the data which is currently stored in `r.Data`. Meaning that the maps will reflect the data in the current reconstruction state. The current state is thereby determined by the recon flags (`r.Parameter.ReconFlags`), the scan parameter (`r.Parameter.Scan`) and the encoding parameter (`r.Parameter.Encoding`).

To calculate the sensitivities during the reconstruction and reconstruct an unfolded image execute the following steps:

- 1) Create a reconstruction object and perform the reconstruction up to the point where you want to unfold the images:

```
>> r = MRecon('senseR2.lab');
>> r.Parameter.Recon.ImageSpaceZeroFill = 'no';
>> r.Parameter.Recon.RemovePOversampling = 'no';
>> r.Parameter.Recon.CoilCombination = 'no';
>> r.Perform;
>> r.ShowData;
```



`r.Data` now contains the uncombined folded coil images

- 2) Create a sensitivity object by calling `MRsense` with the SENSE reference scan and the current reconstruction object `r`:

Release 3 and earlier:

```
>> S = MRsense( ref, r );
```

Release 4 and higher:

```
>> S = MRsense( ref, r, coilsurvey );
```

- 3) Reformat the reference scan to the geometry of the current data and calculate the sensitivity maps:

```
>> S.Perform
```

- 4) Pass the sensitivities to the reconstruction object:

```
>> r.Parameter.Recon.Sensitivities = S;
```

- 5) Unfold the images:

```
>> r.SENSEUnfold;  
>> r.ShowData;
```



## The unfolding function (SENSEUnfold)

The SENSE unfolding process is performed by the MRecon function SENSEUnfold. Whenever it is called it checks if coil sensitivities are available in Parameter.Recon.Sensitivities, and if so, performs a SENSE/CLEAR reconstruction on the data. For a correct SENSE unfolding, the following prerequisites have to be fulfilled before calling SENSEUnfold:

- The data is in image space
- Cartesian sampling in the undersampled direction
- The oversampling in the undersampled direction is not removed
- Image space zero filling has not been performed yet
- The coils are not combined yet



- Parameter.Scan.SENSEFactor is filled

The current implementation of the MRecon Perform function is shown below, where the SENSEUnfold function is called essentially just after the Fourier transformation:

```
MR.ReadData;
MR.DcOffsetCorrection;
MR.PDACorrection;
MR.RandomPhaseCorrection;
MR.MeasPhaseCorrection;
MR.RemoveOversampling;
MR.SortData;
MR.PartialFourier;
MR.GridData;
MR.RingingFilter;
MR.ZeroFill;
MR.K2IM;
MR.EPIPhaseCorrection;
MR.K2IP;
MR.GridderNormalization;
MR.SENSEUnfold;
MR.RemoveOversampling;
MR.CombineCoils;
MR.ZeroFill;
MR.RotateImage;
```

The following alternations in the reconstruction process are possible:

- 1) The RotateImage function can be called before SENSEUnfold
- 2) The oversampling in readout direction can be kept

**However it is crucial that the sensitivity maps are in the same reconstruction state as the unfolded data when SENSEUnfold is called (the sensitivity maps and the data must match!).**

For that purpose, the Rotate and RemoveMOversampling option can be altered in the MRsense class, as described in the next section.

## The MRsense class

There are several options in MRsense which can be set prior to the call of the Perform function:

### Options (default is bold):

#### *CalculateSensitivity [0/1]*

Specifies if the coil sensitivity maps are calculated when the Perform function is called. If set to 0 the refscan is only reformatted but the Sensitivity maps are not calculated (only S.ReformattedCoilData and S.ReformattedBodycoilData is filled). This is useful if one wants to calculate the maps itself.

#### *Mask [0/1]*

Specifies if the raw sensitivity maps should be masked. The raw sensitivity maps are calculated by dividing the individual coil images by the bodycoil image. These raw maps however can be noisy or feature spikes. The masking tries to identify noise only regions and masks them out.

#### *Smooth [0/1]*

Specifies if the raw sensitivity map should be smoothed. The raw sensitivity maps are calculated by dividing the individual coil images by the bodycoil image. These raw maps however can be noisy or feature spikes which are removed by the smoothing process. Smoothing is essentially a linear polynomial fit of the neighborhood in every pixel.

#### *Extrapolate [0/1]*

Specifies if the sensitivity maps should be extrapolated. The raw sensitivity maps are calculated on a low resolution reference scan and used in a (usually) high resolution SENSE scan. Furthermore prior to the calculation of the raw maps, the refscan is masked to exclude noise only regions. In some cases there is a slight mismatch between sensitivity maps and high resolution data due to motion or partial volume effects, which can result in a cropped reconstructed image in areas where the maps are masked out. For this reason the sensitivity maps are usually extrapolated slightly over the borders of the mask.

#### *MatchTargedSize [0/1]*

If set to 1 the outputted sensitivity maps have the same matrix size as the high resolution SENSE scan.

#### *RemoveMOversampling [0/1]*

Specifies if the oversampling in readout direction is removed or not when calling the SENSEUnfold function. If set to zero the oversampling will not be removed in the sensitivity maps.

#### *Rotate [0/1]*

Specifies if the Rotate function is called before or after SENSEUnfold in the reconstruction process. If set to 1 then the sensitivity maps are rotated.

## **Other attributes**

#### *RefScan*

The SENSE reference scan (MRecon object).

#### *TargetScan*

The high resolution SENSE scan (MRecon object). The reference scan is reformatted to the geometry of this scan when the Perform function is called.

#### *CoilData*

The individual coil images of the reference scan (not reformatted)

#### *BodycoilData*

The bodycoil images of the reference scan (not reformatted)

#### *ReformattedCoilData*

The reformatted individual coil images (in the geometry of the high resolution scan)

### *ReformattedBodycoilData*

The reformatted bodycoil images (in the geometry of the high resolution scan)

### *Sensitivity*

The reformatted sensitivity maps (in the geometry of the high resolution scan)

## **Limitations**

- Only Cartesian SENSE is implemented at the moment