



# Chapter 34

## Denoising for Improved Parametric MRI of the Kidney: Protocol for Nonlocal Means Filtering

Ludger Starke, Karsten Tabelow, Thoralf Niendorf, and Andreas Pohlmann

### Abstract

In order to tackle the challenges caused by the variability in estimated MRI parameters (e.g.,  $T_2^*$  and  $T_2$ ) due to low SNR a number of strategies can be followed. One approach is postprocessing of the acquired data with a filter. The basic idea is that MR images possess a local spatial structure that is characterized by equal, or at least similar, noise-free signal values in vicinities of a location. Then, local averaging of the signal reduces the noise component of the signal. In contrast, nonlocal means filtering defines the weights for averaging not only within the local vicinity, but it compares the image intensities between *all* voxels to define “nonlocal” weights. Furthermore, it generally compares not only single-voxel intensities but small spatial patches of the data to better account for extended similar patterns. Here we describe how to use an open source NLM filter tool to denoise 2D MR image series of the kidney used for parametric mapping of the relaxation times  $T_2^*$  and  $T_2$ .

This chapter is based upon work from the COST Action PARENCHIMA, a community-driven network funded by the European Cooperation in Science and Technology (COST) program of the European Union, which aims to improve the reproducibility and standardization of renal MRI biomarkers.

**Key words** Magnetic resonance imaging (MRI), Kidney, Mice, Rats, Parametric mapping,  $T_2$ ,  $T_2^*$ , Diffusion

---

## 1 Introduction

Mapping of the transverse relaxation times  $T_2^*$  and  $T_2$  (or relaxation rates  $R_2^* = 1/T_2^*$  and  $R_2 = 1/T_2$ ) requires series of MR images with different echo times TE: The maps are obtained by fitting the exponential model curve  $S(\text{TE}) = S_0 \exp(-\text{TE}/T_2^{(*)})$  to the signal intensities of each image pixel with increasing TE. This approach is inherently associated with decreasing signal-to-noise ratio (SNR) for the image volumes obtained for longer echo times. Similarly, there is a substantial signal attenuation and SNR reduction in diffusion-weighted MRI versus images acquired without diffusion-sensitizing gradients. Low SNR basically poses two

general problems: First, it leads to a variability for the parameter estimates in the quantitative maps or derived model parameters. Second, it induces a systematic bias for the estimated parameter maps [1]. Please *see* Polzehl and Tabelow [2] for an elaboration of the issues and possible solutions.

In order to tackle the challenges caused by the variability in the estimated parameters due to low SNR a number of strategies can be followed. The obvious one is hardware related and includes improvements on the MR scanner side, such as dedicated RF coils or the use of cryogenically cooled RF probes [3]. Then, SNR is increased by reducing the variability of the measurement. Since the magnitude of the MR signal is directly related to the voxel volume [4], SNR increase can be achieved by reducing the spatial resolution of the image acquisition. This is obviously a very unfavorable solution. Alternatively, signal variability can be reduced by averaging of the signal over multiple acquisitions. This approach comes at the cost of prolonged acquisition times. Acceleration techniques like parallel imaging [5, 6] or compressed sensing [7] allow for more signal averaging within the same time span. Image postprocessing provides another alternative for reducing the variability of the data. Within this chapter, we specifically consider a class of methods that post-processes the acquired data, that is, smoothing. The basic idea is that MR images (like all images) possess a local spatial structure that is characterized by equal, or at least similar, noise-free signal values in vicinities of a location. Then, local averaging of the signal reduces the noise component of the signal. The simplest approach is the application of a Gaussian filter which performs a local averaging with weights defined by a (Gaussian) kernel function and a bandwidth. Due to its simple mathematical formulation it can be performed very fast on data volumes independent of the spatial dimensionality of the data. However, it comes with the blurring of the images, which is evident at edges between different tissues, especially hindering the analysis of fine structures within the volumes.

Alternatively, adaptive smoothing methods have been developed that aim to take the local properties of the specific data at hand into account. The specific methods are based on different methodologies, like anisotropic diffusion [8–11], nonlocal means [12], penalization techniques [13, 14] wavelet filtering [15], model-based methods [16, 17], the propagation-separation approach [18, 19], or other local techniques [20, 21]. Many of these techniques have been applied to diffusion-weighted MRI data but are also applicable for relaxometry measurements. Depending on the specific method and the input data at adaptive smoothing is performed on each image volume separately or uses the combined information from all available raw data. In contrast to nonadaptive filters such as the Gaussian filter, these methods typically require longer computation times and the choice of one or more

smoothing parameters. Furthermore, in a similar way as the Gaussian filter induced blurring of images, adaptive filters might impose artifacts depending on their assumptions on the data like a cartoonesque appearance due to a step-function reconstruction of the images.

Here, we specifically consider the nonlocal means (NLM) filter [22, 23], which is of proven value for MRI. In contrast to defining the weights for averaging within the local vicinity of a location depending on the spatial distance only, it compares the image intensities between *all* voxels to define “nonlocal” weights for averaging intensities. Furthermore, it generally compares not only single-voxel intensities but small spatial patches of the data to better account for extended similar patterns. It is related to the bilateral filtering [24], where the *local* weights are refined by a factor related to local intensity differences; nonlocal means can thus be considered as a bilateral filter with infinite spatial bandwidth making it nonlocal. The filter can be applied to images in two dimensions but also in 3D or for multispectral [25] such as multiecho imaging techniques used in MR relaxometry.

Advantages of the application of the NLM filter are manifold: As a purely postprocessing method it does not require special hardware or sophisticated image acquisition techniques but can be applied offline. Many implementations and extensions of the original algorithm as open-source software are freely available. Examples are provided under: <https://sites.google.com/site/pierriercoupe/softwares/denoising-for-medical-imaging/dwi-denoising/dwi-denoising-software>. Most of the implementations require MATLAB. Although smoothing parameters need to be tuned, default settings provide a very good starting point. Images processed with the NLM filter exhibit an improved SNR, show no extra blurring of the edges and preserve effective spatial resolution. Occasional and slight introduction of large-scale structures along the coordinate axis is a recognized limitation of the NLM filter because the comparison of spatial patches is in favor of these directions.

In this protocol we describe the application of a purpose-build implementation of the NLM filter for 2D MRI relaxometry data (*see Note 1* for implementation details). The open source tool comes in two versions: (1) each echo time is filtered independently (*2D-NLM*) (2) similarities between image patches are estimated jointly for the complete stack of echo images (*stackNLM*). While the first is a straightforward block-wise implementation of the NLM filter, the second proposes a novel method exploiting the redundancy of information in MR relaxometry data (*see Note 1* for implementation details). Both filters reduce the noise level in low SNR relaxometry data with version 2 offering slightly superior results (*see Note 2* for a detailed example). Additionally, application of the MATLAB Image Processing toolbox function `imnlmfilt` is

described. This function performs similar to 2D-NLM while being considerably faster. To complete the preprocessing of MRI relaxometry data, correction of the Rician noise bias is described.

This chapter is part of the book Pohlmann A, Niendorf T (eds) (2020) *Preclinical MRI of the Kidney—Methods and Protocols*. Springer, New York.

---

## 2 Materials

### 2.1 Software Requirements

The processing steps outlined in this protocol require the following:

1. The software development environments *MATLAB*<sup>®</sup> (The MathWorks, Natick, Massachusetts, USA; [mathworks.com](http://mathworks.com)) or *Octave* ([gnu.org/software/octave](http://gnu.org/software/octave)).
2. The MATLAB/Octave implementation of the NLM filter downloadable from [github.com/LudgerS/MRIrelaxometryNLM](https://github.com/LudgerS/MRIrelaxometryNLM)
3. Optional: The MATLAB Image Processing Toolbox release R2018b or newer.
4. A MATLAB/Octave tool for noise bias correction downloadable from [github.com/LudgerS/MRInoiseBiasCorrection](https://github.com/LudgerS/MRInoiseBiasCorrection). This tool includes detailed documentation to facilitate adaptation for other software development environments or use with multichannel coil data.
5. A tool for data import. This tool will depend on the data format:
  - DICOM data—MATLAB contains a built-in function to import DICOM data: `dicomread`. For Octave support is offered by the following package [octave.sourceforge.io/dicom/](https://octave.sourceforge.io/dicom/).
  - Bruker data—Bruker also offers a toolbox to import data into MATLAB directly. To obtain Bruker's *pvttools* contact Bruker software support ([mri-software-support@bruker.com](mailto:mri-software-support@bruker.com)).

### 2.2 Data Requirements

1. The following protocol assumes that a pure noise scan was acquired to reliably estimate the data noise level.
2. See **Note 3** for comments on noise scan acquisition.

---

## 3 Methods

### 3.1 Data Import

These steps should be executed for both the relaxometry data and the noise scan data. In the following sections it is assumed that the relaxometry data was stored in variable `relaxData` and the noise scan in variable `noiseData`.

### 3.1.1 DICOM Data

1. To import dicom data execute the command

```
data = dicomread(filePath);
```

where 'filePath' is a string containing the relative or absolute path of the .dcm file.

### 3.1.2 Bruker Data

For a Linux or MacOS system, substitute the file separator \ by /.

1. Ensure that pvtools is on the search path:

```
addpath(genpath(...\pvtools))
```

where '..\pvtools' specifies the path of the pvtools folder.

2. Store the path to the scan you want to load in variable scanFolder.
3. Read the visu\_pars parameter file:

```
visuPars = readBrukerParamFile([scanFolder, '\visu_pars']);
```

4. Load the Paravision reconstruction:

```
[data, ~] = readBruker2dseq([scanFolder, '\pdata\1\2dseq'], visuPars);
```

## 3.2 Correct Data Scaling

The data scaling needs to be corrected in some cases depending on the data format and vendor if the relaxometry data and noise scan were acquired with different numbers of averages. For example, in Bruker data, the individual averages are simply added instead of computing the mean (*see Note 4* for a procedure to test the scaling convention). Assuming the noise scan was acquired with a single average and the relaxometry data with NA averages, use the following processing step:

1.  $\text{relaxData} = \text{relaxData}/\text{NA}$

## 3.3 Noise Level Estimation

We assume the availability of a pure noise scan. In this case the complete noise data constitutes a background region. Assuming that the noise scan was acquired with a single average and the relaxometry data with NA averages, the noise level can be computed as

1.  $\text{sigma} = \text{std}(\text{noiseData}(:))/(0.6551*\text{sqrt}(\text{NA}))$ ;

The factor 0.6551 accounts for the different standard deviations in background and high SNR regions in MR magnitude images [1, 26, 27]. For data from multichannel coils *see Note 5*.

### 3.4 Application of the NLM Filter

#### 3.4.1 Filtering of Individual Echoes with 2D-NLM

To apply the provided NLM filter implementation, use a loop over all echo times. It is assumed that the relaxometry images are two dimensional and that the echoes are stored along the third dimension of `relaxData`.

1. Set the parameters of the filter (*see Note 6* for an explanation of choices)

```
params.centerDistance = 1; % distance between
individual blocks

params.blockRadius = 1; % size of the blocks;

params.searchRadius = 10; % determines the number
of searched blocks

params.beta = 0.5; % adjusts the filter strength
```

2. Preallocate memory for the filtered data

```
filteredData = zeros(size(relaxData));
```

3. Apply the filter to every echo individually

```
for ii = 1:size(relaxData, 3)

    filteredData(:, :, ii) = nlmFilter2D(relaxData
(:, :, ii), sigma, params);

end
```

#### 3.4.2 Filtering Using `stackNLM`

1. Set the parameters of the filter (*see Note 4* for an explanation of choices)

```
params.centerDistance = 1; % distance between
individual blocks

params.blockRadius = 1; % size of the blocks;

params.searchRadius = 10; % determines the number
of searched blocks

params.beta = 0.3; % adjusts the filter strength
```

2. Directly apply the filter to all echoes

```
filteredData(:, :, ii) = stackNlmFilter(relax-
Data, sigma, params);
```

#### 3.4.3 Filtering Using `MATLAB's imnlmfilt`

`MATLAB's imnlmfilt` works on 2D images. Again a loop over the echo times is used:

1. Preallocate memory for the filtered data

```
filteredData = zeros(size(relaxData));
```

## 2. Apply the filter to every echo individually

```
for ii = 1:size(relaxData, 3)
    filteredData(:, :, ii) = imnlmfilt(relaxData(:, :, ii),
    degreeOfSmoothing, sigma);
end
```

**3.5 Bias Correction**

To correct the Rician noise bias, the MRInoiseBiasCorrection toolbox is used. Importantly, the noise level  $\sigma$  of the original data and not that of the filtered image needs to be given as an input parameter.

## 1. Ensure that the toolbox is on the search path

```
addpath(...\MRInoiseBiasCorrection)
```

where ‘...\`MRInoiseBiasCorrection`’ specifies the path of the toolbox folder.

## 2. Apply the noise bias correction

```
correctedImage = correctNoiseBias(imageData,
sigma, 1);
```

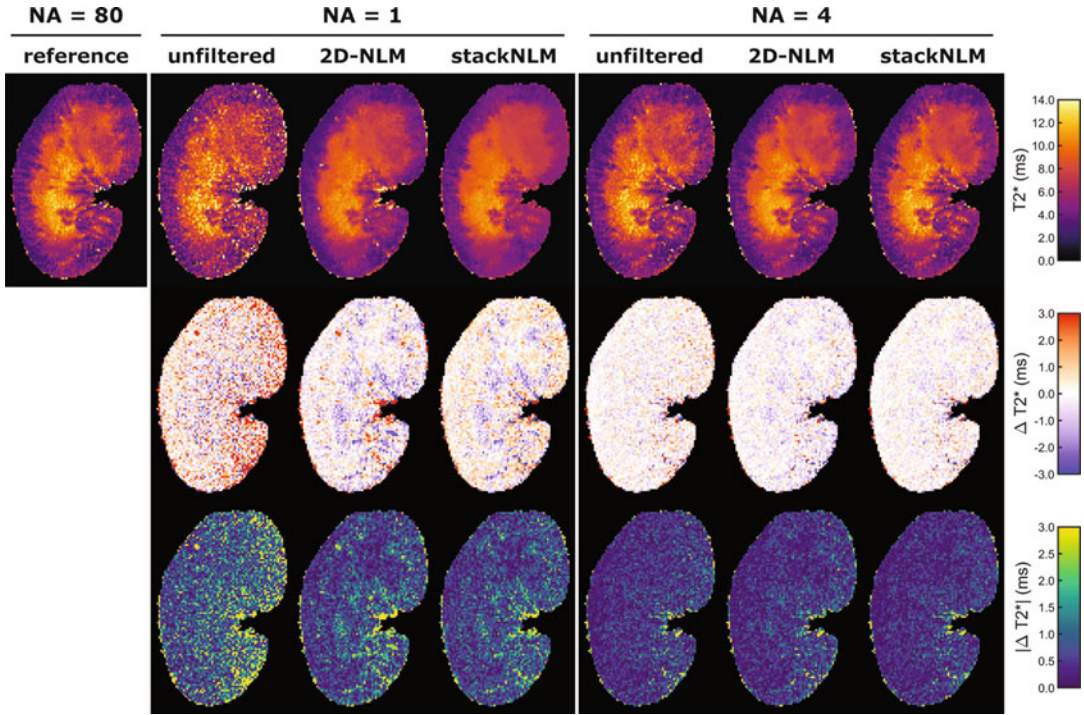
The last argument specifies the number of channels in the RF coil.

---

**4 Notes**

1. The provided tool follows the block-wise implementation of the NLM filter outlined by Coupé et al. for 3D data [28]. The noise level  $\sigma$  is expected to be provided by separate means such as the noise scan described in this protocol. The filtering parameter  $h$  is set to  $\sqrt{2\beta\sigma^2n}$ , where  $n$  is the number of pixels in the compared blocks and  $\beta$  a parameter which can be controlled by the user (*see Note 5*). Noise bias correction was not integrated into the filter but is recommended as a subsequent step.
2. We present an example application to showcase the potential of the NLM filter for  $T_2^*$  relaxometry. For this purpose the kidney of an ex vivo rat phantom was scanned with a multiecho gradient-echo (MGE) sequence (TR = 50 ms, TE = 2.14 ms, 2.47 ms echo spacing, ten echoes,  $[38.2 \times 48.5]$  mm<sup>2</sup> FOV, 1 mm slice thickness,  $[202 \times 256]$  image matrix, 80 repetitions, one average per repetition). The average of all 80 repetitions was used as a reference measurement.  $T_2^*$  maps were computed from unfiltered data and data filtered following the protocols of Subheading 3.4.1 (2D-NLM) and Subheading 3.4.2 (stackNLM).

Figure 1 shows maps for a single repetition (NA = 1), the average of four repetitions (NA = 4) and the reference data

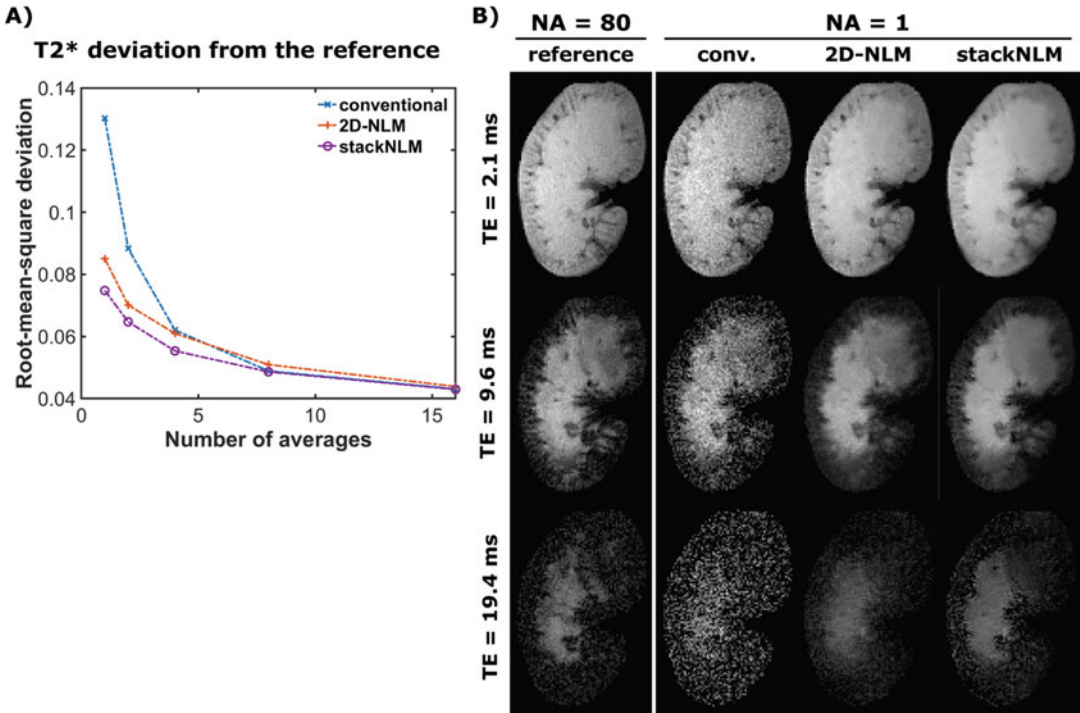


**Fig. 1** Filtered and unfiltered kidney  $T_2^*$  maps acquired in an ex vivo rat phantom. The reference was computed from unfiltered data and NA denotes the number of averages. The second row shows the  $T_2^*$  deviation from the reference while the third row shows the magnitude of the  $T_2^*$  deviation

with 80 averages. Maps showing the deviation from the reference ( $\Delta T_2^*$ ) and the magnitude deviation ( $|\Delta T_2^*|$ ) highlight the strongly reduced noise level in filtered images for NA = 1. No bias is introduced and edges are preserved. The filter working on the complete stack of echoes (stackNLM) is more robust regarding outliers. For NA = 4 only very little improvement is achieved due to the already good data quality, yet also no artifacts are introduced.

The root-mean-square deviations (RMSD) of the  $T_2^*$  maps from the reference support this visual assessment (Fig. 2a). At low SNR (low number of averages) the filtered data outperform the maps computed without filtering. The improvement exceeds the gain achieved by doubling the measurement time. At higher SNR filtered and unfiltered data perform similarly, stackNLM consistently performs better than 2D-NLM. Figure 2b shows selected echoes for NA = 1 and the reference. Here the improved performance of stackNLM becomes obvious for data acquired with TE = 19.4 ms where signal is recovered with sharper detail.

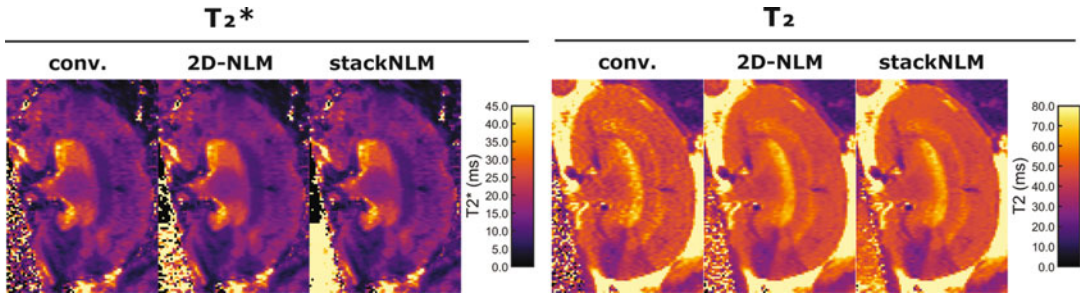




**Fig. 2** (a) Root-mean-square deviation (RMSD) from the reference (NA = 80, unfiltered) for  $T_2^*$  maps computed from filtered and unfiltered data. (b) Image data for selected echo times (TE). The scaling is set individually for each row

Figure 3 shows an in vivo example for both  $T_2^*$  and  $T_2$  data. Here little filtering is performed for the high SNR  $T_2^*$  data, whereas the  $T_2$  map is effectively denoised.

3. A pure noise scan is acquired by setting the excitation flip angle and reference power to zero so that no excitation occurs and pure noise is acquired. The receiver gain needs to be set identical to all other scans for which the noise level should be determined. The number of averages can be reduced to one, as the resulting change in noise level is easily compensated. Alternatively, the output of the RF power supply can be disconnected following the system adjustments.
4. To determine the employed convention of averaging, acquire high SNR phantom data with varying numbers of averages while keeping all other parameters fixed. Import the data as described in Subheading 3.1. If all scans show the same signal magnitude, the scaling step described in Subheading 3.2 should be omitted. However if for example doubling the number of averages also doubles the signal amplitude, rescaling should be performed.



**Fig. 3** Filtered and unfiltered  $T_2$  and  $T_2^*$  maps acquired in a rat in vivo. In this experimental set-up (a healthy naive rat kidney being imaged using a dedicated 9.4T small animal MR system) the SNR is sufficiently high for the denoising to have only minor denoising effect. The parameter  $\beta$  was set to 0.4 for the stackNLM filter to increase the smoothing effect

5. Assuming a sum-of-squares reconstruction, the factor 0.6551 should be replaced by 0.6824, 0.6953, 0.7014 or 0.7043 for 2, 4, 8 or 16 receive element coil data [29].
6. The block-wise NLM filter compares overlapping square image patches or blocks. The larger the distance between block centers (params.centerDistance), the fewer blocks are in the image. Reducing the number of blocks will accelerate the filter but might lead to artifacts in the vicinity of contrast edges. params.blockRadius determines the size of the blocks and must always be larger than params.centerDistance to ensure overlap. The block is a square patch of area  $(\text{params.centerDistance} + 1)^2$ . Increasing the size of blocks will improve performance in homogeneous areas but prevent the filter from finding similar blocks in areas with small features. params.searchRadius specifies the number of compared image blocks: each block is compared with the adjacent  $(\text{params.searchRadius} + 1)^2$  blocks to determine the corresponding filter weights. In the ideal case the algorithm would always search the complete image for similar blocks; however, the computational cost to do so would be prohibitive while the benefit diminishes fast with increasing search radius. params.beta specifies the parameter  $\beta$  introduced in **Note 1**. While theoretical considerations suggest a value close to 1, the optimal value of  $\beta$  is known to vary with the noise level [28]. Increasing the value of  $\beta$  will increase the smoothing effect. The proposed values of 0.5 for 2D-NLM and 0.3 for stackNLM were found to perform well over a wide range of noise levels.

## Acknowledgments

This work was funded, in part (S.W. and A.P.; DFG WA2804, DFG PO1869), by the Deutsche Forschungsgemeinschaft and, in part (Thoralf Niendorf and Andreas Pohlmann), by the German Research Foundation (Gefördert durch die Deutsche Forschungsgemeinschaft (DFG), Projektnummer 394046635, SFB 1365, RENOPROTECTION). Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project number 394046635, SFB 1365, RENOPROTECTION). This chapter is based upon work from COST Action PARENCHIMA, supported by European Cooperation in Science and Technology (COST). COST ([www.cost.eu](http://www.cost.eu)) is a funding agency for research and innovation networks. COST Actions help connect research initiatives across Europe and enable scientists to enrich their ideas by sharing them with their peers. This boosts their research, career, and innovation.

PARENCHIMA ([renalMRI.org](http://renalMRI.org)) is a community-driven Action in the COST program of the European Union, which unites more than 200 experts in renal MRI from 30 countries with the aim to improve the reproducibility and standardization of renal MRI biomarkers.

## References

1. Gudbjartsson H, Patz S (1995) The Rician distribution of noisy MRI data. *Magn Reson Med* 34:910–914
2. Polzehl J, Tabelow K (2016) Low SNR in diffusion MRI models. *J Am Stat Assoc* 111 (516):1480–1490
3. Niendorf T, Pohlmann A, Reimann HM et al (2015) Advancing cardiovascular, neurovascular, and renal magnetic resonance imaging in small rodents using cryogenic radiofrequency coil technology. *Front Pharmacol* 6:255
4. Edelstein W, Glover G, Hardy C, Redington R (1986) The intrinsic signal-to-noise ratio in NMR imaging. *Magn Reson Med* 3 (4):604–618
5. Pruessmann KP, Weiger M et al (1999) SENSE: sensitivity encoding for fast MRI. *Magn Reson Med* 42(5):952–962
6. Griswold MA, Jakob PM et al (2002) Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magn Reson Med* 47 (6):1202–1210
7. Lustig M, Donoho D, Pauly JM (2007) Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn Reson Med* 58:1182–1195
8. Ding Z, Gore J, Anderson A (2005) Reduction of noise in diffusion tensor images using anisotropic smoothing. *Magn Reson Med* 53 (2):485–490
9. Parker G, Schnabel J, Symms M, Werring D, Barker G (2000) Nonlinear smoothing for reduction of systematic and random errors in diffusion tensor imaging. *J Magn Reson Imaging* 11:702–710
10. Xu Q, Anderson AW, Gore JC, Ding Z (2010) Efficient anisotropic filtering of diffusion tensor images. *Magn Reson Imaging* 28 (2):200–211
11. Duits R, Franken E (2011) Left-invariant diffeomorphisms on the space of positions and orientations and their application to crossing-preserving smoothing of HARDI images. *Int J Comput Vis* 92(3):231–264
12. Manjón JV, Thacker NA, Lull JJ, Garcia-Martí G, Martí-Bonmati L, Robles M (2009) Multicomponent MR image denoising. *Int J Biomed Imaging* 2009:756897
13. McGraw T, Vemuri B, Özarlan E, Chen Y, Mareci T (2009) Variational denoising of diffusion weighted MRI. *Inverse Probl Imaging* 3:625–648

14. Haldar JP (2014) Low-rank modeling of local k-space neighborhoods (LORAKS) for constrained MRI. *IEEE Trans Med Imaging* 33 (3):668–681
15. Lohmann G, Bohn S, Müller K, Trampel R, Turner R (2010) Image restoration and spatial resolution in 7-Tesla magnetic resonance imaging. *Magn Reson Med* 64(1):15–22
16. Fletcher PT (2004) Statistical variability in nonlinear spaces: application to shape analysis and DT-MRI, Ph.D. thesis. University of North Carolina at Chapel Hill, Chapel Hill, NC
17. Tabelow K, Polzehl J, Spokoiny V, Voss HU (2008) Diffusion tensor imaging: structural adaptive smoothing. *Neuroimage* 39:1763–1773
18. Becker SMA, Tabelow K, Voss HU, Anwender A, Heidemann RM, Polzehl J (2012) Position-orientation adaptive smoothing of diffusion weighted magnetic resonance data (POAS). *Med Image Anal* 16 (6):1142–1155
19. Becker SMA, Tabelow K, Mohammadi S, Weiskopf N, Polzehl J (2014) Adaptive smoothing of multi-shell diffusion-weighted magnetic resonance data by msPOAS. *Neuroimage* 95:90–105
20. Aja-Fernández S, Niethammer M, Kubicki M, Shenton ME, Westin C-F (2008) Restoration of DWI data using a Rician LMMSE estimator. *IEEE Trans Med Imaging* 27:1389–1403
21. Tristán-Vega A, Aja-Fernández S (2010) DWI filtering using joint information for DTI and HARDI. *Med Image Anal* 14(2):205–218
22. Buades A, Coll B, Morel J (2005) A non-local algorithm for image denoising. In: *IEEE Computer Society conference on computer vision and pattern recognition (CVPR'05)*, San Diego, CA, vol 2. IEEE Computer Society, Washington, DC, pp 60–65
23. Wiest-Daesslé N, Prima S, Coupe P, Morrissey SP, Barillot C (2007) Nonlocal means variants for denoising of diffusion-weighted and diffusion tensor MRI. *MICCAI* 10(2):344–351
24. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: *Proceedings of the sixth international conference on computer vision (ICCV '98)*. IEEE Computer Society, Washington, DC
25. Bouhrara M, Bonny J-M, Ashinsky BG, Maring MC, Spencer RG (2017) Noise estimation and reduction in multispectral magnetic resonance images. *IEEE Trans Med Imaging* 36 (1):181–193
26. National Electrical Manufacturers Association (2001) Determination of signal-to-noise ratio (SNR) in diagnostic magnetic resonance imaging. NEMA standards publication MS 1–200. NEMA, Arlington, VA
27. Henkelman RM (1985) Measurement of signal intensities in the presence of noise in MR images. *Med Phys* 12(2):232–233
28. Coupe P, Yger P, Prima S, Hellier P, Kervrann C, Barillot C (2008) An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE Trans Med Imaging* 27(4):425–441
29. Constantinides CD, Atalar E, McVeigh ER (1997) Signal-to-noise measurements in magnitude images from NMR phased arrays. *Magn Reson Med* 38(5):852–857

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

