Supplementary file for "Model-based super-resolution reconstruction with joint motion estimation for improved quantitative MRI parameter mapping"

Quinten Beirinckx[a,d,*], Ben Jeurissen[a,d], Michele Nicastro[a,d], Dirk H. J. Poot[b], Marleen Verhoye[c,d], Arnold J. den Dekker[a,d] and Jan Sijbers[a,d]

[a] *imec-Vision Lab, Department of Physics, University of Antwerp, Antwerp, Belgium*
[b] *Biomedical Imaging Group Rotterdam, Department of Radiology and Nuclear Medicine, Erasmus MC, Rotterdam, The Netherlands*
[c] *Bio-Imaging Lab, Department of Biomedical Sciences, University of Antwerp, Antwerp, Belgium*
[d] *μNEURO Research Centre of Excellence, University of Antwerp, Antwerp, Belgium*

ARTICLE INFO

Keywords:
Bayesian estimation
model-based reconstruction
motion correction
quantitative magnetic resonance imaging
super-resolution

ABSTRACT

This document includes supplementary material that complements the main body of the paper. In Section 1, we provide analytical derivations of the Jacobian and Hessian of the alternating minimization scheme which was used to obtain the joint Maximum a Posteriori estimates of the tissue and motion parameters. Next, Section 2 elaborates on the implementation of the forward model operators and the computational requirements. Finally, a section has been added to explain the choice of a realistic motion parameter set for the whole brain simulation experiments, to provide confidence intervals for the quantitative performance measures, to elaborate on the spatial resolution assessment by means of edge profile fitting, and to illustrate the echo time selection for the *in vivo* T_2 mapping experiment.

1. Analytical derivatives for optimization of problems (P.1) and (P.2)

The proposed joint MAP estimation consists of the following iterative recursive procedure:

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\tilde{\boldsymbol{s}}}(\hat{\boldsymbol{\vartheta}}^{(t)}, \boldsymbol{\theta}|\tilde{\boldsymbol{s}}) \tag{P.1}$$

$$\hat{\boldsymbol{\vartheta}}^{(t+1)} = \arg\min_{\boldsymbol{\vartheta}} \left[\mathcal{L}_{\tilde{\boldsymbol{s}}}(\boldsymbol{\vartheta}, \hat{\boldsymbol{\theta}}^{(t+1)}|\tilde{\boldsymbol{s}}) + \sum_{q=1}^{Q} \frac{2}{\lambda_q} \mathrm{TV}(\boldsymbol{\vartheta}_q) \right] \tag{P.2}$$

with

$$\mathcal{L}_{\tilde{\boldsymbol{s}}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}|\tilde{\boldsymbol{s}}) = \sum_{n=1}^{N} \mathcal{L}_{\tilde{\boldsymbol{s}}_n}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n|\tilde{\boldsymbol{s}}_n) = -\sum_{n=1}^{N} \log p(\tilde{\boldsymbol{s}}_n|\boldsymbol{\vartheta}, \boldsymbol{\theta}_n) \tag{S1}$$

where the summation runs over all N contrast-weighted low-resolution (LR) images $\tilde{\boldsymbol{s}}_n$. Problems (P.1) and (P.2) are minimized using a trust-region Newton method (Coleman and Li, 1994). Such a gradient-based optimization algorithm benefits from having analytical expressions for the Jacobian and Hessian to avoid time-consuming finite difference computations. These analytical expressions are derived hereafter.

Nomenclature In what follows, we rewrite the forward operator sequence as $\boldsymbol{A}_n = \boldsymbol{D}\boldsymbol{B}\boldsymbol{G}_n$ and its adjoint sequence as $\boldsymbol{A}_n^T = \boldsymbol{G}_n^T \boldsymbol{B}^T \boldsymbol{D}^T$ to ease the notation, unless stated otherwise. As such, the forward model introduced in (1) of section 2.1 in the main body of the paper, can be written more concisely as:

$$\boldsymbol{s}_n = \left| \boldsymbol{D}\boldsymbol{B}\boldsymbol{G}_n \boldsymbol{M}_{\theta_n} \boldsymbol{r}_n \right| = \left| \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} \boldsymbol{r}_n \right|. \tag{S2}$$

The authors gratefully acknowledge support from Belgian Science Policy Prodex (Grant ISLRA 2009–1062), the Research Foundation Flanders (FWO) through project funding G084217N and 12M3119N, and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 764513.

[*] Corresponding author, Address: Vision Lab, University of Antwerp (CDE), Universiteitsplein 1, N.1.16, B-2610 Antwerpen, Belgium
quinten.beirinckx@uantwerpen.be (Quinten Beirinckx)
ORCID(s): 0000-0002-5339-052X (Quinten Beirinckx)

1.1. MAP estimation of motion parameters

Assuming no dependence of $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$ through index n, the rigid inter-image motion parameter optimization problem (P.1) can be decoupled into N parallel subproblems. In what follows, the optimization of a single rigid motion set $\boldsymbol{\theta}_n$ corresponding with LR image $\tilde{\boldsymbol{s}}_n$ is considered.

The cost function of this estimation problem is given by

$$\begin{aligned}\mathcal{L}_{\tilde{s}_n}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n|\tilde{\boldsymbol{s}}_n) &= -\log p_n(\tilde{\boldsymbol{s}}_n; \boldsymbol{\vartheta}, \boldsymbol{\theta}_n) \\ &= \sum_{l=1}^{N_s}\left[-\log \tilde{s}_{nl} + \log \sigma_{nl}^2 + \frac{\tilde{s}_{nl}^2}{2\sigma_{nl}^2} + \frac{s_{nl}^2(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{2\sigma_{nl}^2} - \log I_0\left(\frac{\tilde{s}_{nl}s_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\sigma_{nl}^2}\right)\right].\end{aligned} \tag{S3}$$

Keeping only terms that are function of the unknown parameter vector $\boldsymbol{\theta}_n$, as only those are relevant for the minimization, (S3) simplifies to

$$\mathcal{L}_{\tilde{s}_n}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n|\tilde{\boldsymbol{s}}_n) \sim \sum_{l=1}^{N_s}\left[\frac{s_{nl}^2(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{2\sigma_{nl}^2} - \log I_0\left(\frac{\tilde{s}_{nl}s_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\sigma_{nl}^2}\right)\right]. \tag{S4}$$

Assuming inter-rigid motion, the motion parameter vector $\boldsymbol{\theta}_n \in \mathbb{R}^{6\times 1}$ is defined as,

$$\boldsymbol{\theta}_n = \{\theta_{nk}\}_{k=1}^{6} = \left[t_{xn}, t_{yn}, t_{zn}, \alpha_n, \beta_n, \gamma_n\right]^T. \tag{S5}$$

We then define the gradient w.r.t. the motion parameter θ_{nk} by taking the respective derivative of (S4):

$$\nabla_{nk}^{\mathcal{L}} = \frac{\partial \mathcal{L}_{\tilde{s}_n}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n|\tilde{\boldsymbol{s}}_n)}{\partial \theta_{nk}} = \boldsymbol{b}_n^T \boldsymbol{c}_{nk} \tag{S6}$$

where

$$\begin{cases} \boldsymbol{b}_n = \dfrac{\partial \mathcal{L}_{\tilde{s}_n}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n|\tilde{\boldsymbol{s}}_n)}{\partial \boldsymbol{s}_n(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)} = \left[\dfrac{\boldsymbol{s}_n(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\boldsymbol{\sigma}_n^2} - \dfrac{\tilde{\boldsymbol{s}}_n}{\boldsymbol{\sigma}_n^2}\dfrac{I_1\left(\dfrac{\tilde{\boldsymbol{s}}_n\boldsymbol{s}_n(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\boldsymbol{\sigma}_n^2}\right)}{I_0\left(\dfrac{\tilde{\boldsymbol{s}}_n\boldsymbol{s}_n(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\boldsymbol{\sigma}_n^2}\right)}\right] \\[2em] \boldsymbol{c}_{nk} = \dfrac{\partial \boldsymbol{s}_n(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\partial \theta_{nk}} = \dfrac{\partial |\boldsymbol{A}_n\boldsymbol{M}_{\theta_n}\boldsymbol{r}_n|}{\partial \theta_{nk}} = \operatorname{sgn}(\boldsymbol{A}_n\boldsymbol{M}_{\theta_n}\boldsymbol{r}_n) \odot \left(\boldsymbol{A}_n\dfrac{\partial \boldsymbol{M}_{\theta_n}}{\partial \theta_{nk}}\boldsymbol{r}_n\right)\end{cases} \tag{S7}$$

with $\boldsymbol{b}_n = \{b_{nl}\}_{l=1}^{N_s} \in \mathbb{R}^{N_s\times 1}$, $\boldsymbol{c}_{nk} \in \mathbb{R}^{N_s\times 1}$, and where $\odot$ stands for point-wise multiplication. Finally, substitution of (S7) in (S6) results in (S8) for $\nabla_{nk}^{\mathcal{L}} \in \mathbb{R}$

$$\nabla_{nk}^{\mathcal{L}} = \underbrace{\boldsymbol{r}_n^T}_{\in\mathbb{R}^{1\times N_r}} \underbrace{\frac{\partial \boldsymbol{M}_{\theta_n}^T}{\partial \theta_{nk}}\boldsymbol{A}_n^T\left[\underbrace{\operatorname{sgn}(\boldsymbol{A}_n\boldsymbol{M}_{\theta_n}\boldsymbol{r}_n)}_{\in\mathbb{R}^{N_s\times 1}} \odot \underbrace{\left(\frac{\boldsymbol{s}_n(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\boldsymbol{\sigma}_n^2} - \frac{\tilde{\boldsymbol{s}}_n}{\boldsymbol{\sigma}_n^2}\frac{I_1\left(\frac{\tilde{\boldsymbol{s}}_n\boldsymbol{s}_n(\boldsymbol{\vartheta},\boldsymbol{\theta}_n)}{\boldsymbol{\sigma}_n^2}\right)}{I_0\left(\frac{\tilde{\boldsymbol{s}}_n\boldsymbol{s}_n(\boldsymbol{\vartheta},\boldsymbol{\theta}_n)}{\boldsymbol{\sigma}_n^2}\right)}\right)}_{\in\mathbb{R}^{N_s\times 1}}\right]}_{\in\mathbb{R}^{N_r\times 1}} \tag{S8}$$

The exact implementation of $\dfrac{\partial \boldsymbol{M}_{\theta_n}^T}{\partial \theta_{nk}}$ and the SRR forward model operators is further discussed in Section 2 hereafter.

In addition, to avoid drift of the coordinate system, a zero-mean motion constraint is used that enforces the geometric mean of the motion parameters $\hat{\boldsymbol{\theta}}^{(t)}$ to be the identity transformation. Specifically, the motion parameters of each LR image are drift corrected after solving problem (P.1) by composing the inverse of the geometric mean of $\hat{\boldsymbol{\theta}}^{(t)}$ to each motion parameter set $\hat{\boldsymbol{\theta}}_n^{(t)}$.

1.2. MAP estimation of tissue parameters

In contrast to problem (P.1), the tissue parameter estimation problem (P.2) is a large-scale minimization problem. The cost function of this estimation problem is given by

$$\mathcal{L}_{\tilde{\boldsymbol{s}}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}|\tilde{\boldsymbol{s}}) + \sum_{q=1}^{Q} \frac{2}{\lambda_q}\mathrm{TV}(\boldsymbol{\vartheta}_q) = -\sum_{n=1}^{N} \log p_{\tilde{\boldsymbol{s}}_n}(\tilde{\boldsymbol{s}}_n; \boldsymbol{\vartheta}, \boldsymbol{\theta}_n) + \sum_{q=1}^{Q} \frac{2}{\lambda_q}\mathrm{TV}(\boldsymbol{\vartheta}_q). \tag{S9}$$

The tissue parameter maps to be inferred are $\boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_q\}_{q=1}^{Q} \in \mathbb{R}^{N_r \times Q}$, with $\boldsymbol{\vartheta}_q = \{\vartheta_{qj}\}_{j=1}^{N_r} \in \mathbb{R}^{N_r \times 1}$ the q^{th} tissue parameter map and $\boldsymbol{\vartheta}_{\bullet j} \in \mathbb{R}^{Q \times 1}$ all tissue parameters of the j^{th} voxel of $\boldsymbol{\vartheta}_q$. The gradient of the cost function w.r.t. the tissue parameter element ϑ_{qj} can be written as:

$$\nabla_{\vartheta_{qj}}^{\mathcal{L}} = \frac{\partial \mathcal{L}_{\tilde{\boldsymbol{s}}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}|\tilde{\boldsymbol{s}})}{\partial \vartheta_{qj}} = \sum_{n=1}^{N}\sum_{l=1}^{N_s} \frac{\partial \mathcal{L}_{\tilde{s}_{nl}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n|\tilde{\boldsymbol{s}}_n)}{\partial \vartheta_{qj}} = \sum_{n=1}^{N}\sum_{l=1}^{N_s} b_{nl}\frac{\partial \boldsymbol{s}_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\partial \vartheta_{qj}} = \sum_{n=1}^{N}\sum_{l=1}^{N_s} b_{nl}\boldsymbol{J}_{nl,qj}. \tag{S10}$$

Here, $\boldsymbol{J}_{nl,qj}$ denotes the elements of the Jacobian matrix, which can be further expressed by

$$\boldsymbol{J}_{nl,qj} = \frac{\partial \boldsymbol{s}_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\partial \vartheta_{qj}} = \frac{\partial \left| \sum_{j=1}^{N_r} \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)\right|}{\partial \vartheta_{qj}} = \mathrm{sgn}\left(\varphi_n\right) \sum_{j=1}^{N_r} \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} \frac{\partial f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{qj}}, \tag{S11}$$

where we write $\varphi_n = \sum_{j=1}^{N_r} \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)$ to ease the notation in what follows.

Furthermore, the upwind Total Variation term $\mathrm{TV}(\boldsymbol{\vartheta}_q)$, as described in section 2.2.4 of the main body of the paper, is given by:

$$\mathrm{TV}(\boldsymbol{\vartheta}_q) = \sum_j \left[\sqrt{\zeta_{qj}} - \epsilon\right] \tag{S12}$$

with

$$\zeta_{qj} = \epsilon^2 + \sum_{m \in \{x,y,z\}} \left[\left(\Delta^{m,+}(\vartheta_{qj})\right)^2 + \left(\Delta^{m,-}(\vartheta_{qj})\right)^2\right]. \tag{S13}$$

The derivative of (S12) w.r.t. element ϑ_{qj} is then given by

$$\frac{\partial \mathrm{TV}(\boldsymbol{\vartheta}_q)}{\partial \vartheta_{qj}} = \frac{1}{2}\sum_j \left(\zeta_{qj}\right)^{-1/2} \frac{\partial \zeta_{qj}}{\partial \vartheta_{qj}}. \tag{S14}$$

Note that a small offset $\epsilon > 0$ is introduced in (S13) to avoid derivative singularities of TV when $\boldsymbol{\vartheta}_q$ is locally constant.

The second order derivatives of cost function $\mathcal{L}_{\tilde{\boldsymbol{s}}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}|\tilde{\boldsymbol{s}})$ w.r.t. the tissue parameter elements ϑ_{qj} can be calculated by taking the derivatives one order higher:

$$H_{\vartheta_{qj}\vartheta_{q'j'}}^{\mathcal{L}} = \frac{\partial}{\partial \vartheta_{qj}}\left(\nabla_{\vartheta_{q'j'}}^{\mathcal{L}}\right) = \sum_{n=1}^{N}\sum_{l=1}^{N_s} \frac{\partial}{\partial \vartheta_{qj}}\left(b_{nl}\boldsymbol{J}_{nl,q'j'}\right) = \sum_{n=1}^{N}\sum_{l=1}^{N_s}\left(\nabla_{\vartheta_{qj}}^{b_{nl}}\boldsymbol{J}_{nl,q'j'} + b_{nl}\nabla_{\vartheta_{qj}}^{\boldsymbol{J}_{nl}}\right). \tag{S15}$$

Using the shorthand notation $z_{nl} = \dfrac{\tilde{s}_{nl} s_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\sigma_{nl}^2}$, the gradient terms $\nabla_{\vartheta_{qj}}^{b_{nl}}$ and $\nabla_{\vartheta_{qj}}^{J}$ are given by

$$
\begin{aligned}
\nabla_{\vartheta_{qj}}^{b_{nl}} &= \frac{\partial b_{nl}}{\partial \vartheta_{qj}} \\
&= \frac{\partial}{\partial \vartheta_{qj}} \left(\frac{\partial \mathcal{L}_{\tilde{s}_{nl}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n | \tilde{\boldsymbol{s}}_n)}{\partial s_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)} \right) \\
&= \frac{\partial^2 \mathcal{L}_{\tilde{s}_{nl}}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n | \tilde{\boldsymbol{s}}_n)}{\partial s_{nl}^2(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)} \frac{\partial s_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\partial \vartheta_{qj}} \\
&= \left[\frac{1}{\sigma_{nl}^2} - \frac{\tilde{s}_{nl}^2}{\sigma_{nl}^4} \left[1 - \frac{1}{z_{nl}} \frac{I_1\left(z_{nl}\right)}{I_0\left(z_{nl}\right)} - \frac{I_1^2\left(z_{nl}\right)}{I_0^2\left(z_{nl}\right)} \right] \right] \boldsymbol{J}_{nl,qj},
\end{aligned}
\tag{S16}
$$

$$
\begin{aligned}
\nabla_{\vartheta_{qj}}^{J_{nl}} &= \frac{\partial \boldsymbol{J}_{nl,q'j'}}{\partial \vartheta_{qj}} \\
&= \frac{\partial}{\partial \vartheta_{qj}} \left(\frac{\partial s_{nl}(\boldsymbol{\vartheta}, \boldsymbol{\theta}_n)}{\partial \vartheta_{q'j'}} \right) \\
&= \frac{\partial}{\partial \vartheta_{qj}} \left(\operatorname{sgn}\left(\varphi_n\right)\right) \sum_{j=1}^{N_{\boldsymbol{r}}} \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} \frac{\partial f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{q'j'}} + \operatorname{sgn}\left(\varphi_n\right) \sum_{j=1}^{N_{\boldsymbol{r}}} \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} \frac{\partial^2 f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{qj} \partial \vartheta_{q'j'}} \\
&= \operatorname{sgn}\left(\varphi_n\right) \sum_{j=1}^{N_{\boldsymbol{r}}} \boldsymbol{A}_n \boldsymbol{M}_{\theta_n} \frac{\partial^2 f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{qj} \partial \vartheta_{q'j'}},
\end{aligned}
\tag{S17}
$$

where we have used that $\dfrac{d \operatorname{sgn}(x)}{dx} = 2\delta(x)$.

The partial derivatives $\frac{\partial f_n(\boldsymbol{\vartheta}_{\bullet j})}{\partial \vartheta_{qj}}$ and $\frac{\partial^2 f_n(\boldsymbol{\vartheta}_{\bullet j})}{\partial \vartheta_{qj} \partial \vartheta_{q'j'}}$ depend on the signal model of choice. In this work, a T1-relaxometry model was adopted as a showcase example (Barral et al., 2010):

$$
f_n(\boldsymbol{\vartheta}_{\bullet j}) = \rho_j \left(1 - 2\, e^{-\frac{\mathrm{TI}_n}{T_{1,j}}} \right),
\tag{S18}
$$

with $\boldsymbol{\vartheta}_{\bullet j} = [\rho_j, T_{1,j}]^T$ the tissue parameter vector at position $\boldsymbol{x}_j$. A more extensive description of this signal model is given in section 2.1 of the main body of the paper. The signal model considers $Q = 2$ tissue parameter maps. Keeping track of the HR voxel index $j = 1, \ldots, N_{\boldsymbol{r}}$, and tissue parameter index $q = 1, \ldots, Q$, the first and second order derivatives of $f_n(\boldsymbol{\vartheta}_{\bullet j})$ w.r.t. the tissue parameters ϑ_{qj} are defined by (S19) and (S20), which are given as

$$
\frac{\partial f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{1j}} = 1 - 2e^{-\frac{\mathrm{TI}_n}{T_{1,j}}}, \qquad \frac{\partial f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{2j}} = -2\rho_j e^{-\frac{\mathrm{TI}_n}{T_{1,j}}} \left(\frac{\mathrm{TI}_n}{\left(T_{1,j}\right)^2} \right)
\tag{S19}
$$

$$
\frac{\partial^2 f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{1j}^2} = 0, \qquad \frac{\partial^2 f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{2j}^2} = -2\rho_j e^{-\frac{\mathrm{TI}_n}{T_{1,j}}} \left(\frac{\mathrm{TI}_n}{\left(T_{1,j}\right)^3} \right) \left(\frac{\mathrm{TI}_n}{T_{1,j}} - 2 \right), \qquad \frac{\partial^2 f_n\left(\boldsymbol{\vartheta}_{\bullet j}\right)}{\partial \vartheta_{1j} \partial \vartheta_{2j'}} = -2e^{-\frac{\mathrm{TI}_n}{T_{1,j}}} \left(\frac{\mathrm{TI}_n}{\left(T_{1,j}\right)^2} \right)
\tag{S20}
$$

Finally, we also give an expression for the second order derivative of the upwind Total Variation prior term in (S12):

$$\begin{aligned}\frac{\partial}{\partial \vartheta_{qj}}\left(\frac{\partial \mathrm{TV}(\boldsymbol{\vartheta}_q)}{\partial \vartheta_{q'j'}}\right) &= \frac{\partial}{\partial \vartheta_{qj}}\left(\frac{1}{2}\sum_j \left(\zeta_{qj}\right)^{-1/2}\frac{\partial \zeta_{qj}}{\partial \vartheta_{q'j'}}\right)\\ &= \frac{1}{2}\sum_j\left[\left(\zeta_{qj}\right)^{-1/2}\frac{\partial^2 \zeta_{qj}}{\partial \vartheta_{qj}\partial \vartheta_{q'j'}} - \frac{1}{2}\left(\zeta_{qj}\right)^{-3/2}\left(\frac{\partial \zeta_{qj}}{\partial \vartheta_{qj}}\right)\left(\frac{\partial \zeta_{qj}}{\partial \vartheta_{q'j'}}\right)\right].\end{aligned} \tag{S21}$$

Please note that for problem (P.2) the Hessian matrix was not explicitly stored in memory, but was implemented as a Hessian multiply function. This function gives the result of a Hessian-times-vector product without computing the Hessian directly, and thus avoids excessive memory usage.

2. SRR forward model operators and computational requirements

2.1. Warping operators and derivatives

The proposed SRR framework uses different warping operators in the forward model, described by (1) in section 2.1. The operator $\boldsymbol{G}_n$ describes the known geometric transformation, extracted from the LR image acquisition header information. This operator models the SRR acquisition, in which multiple LR contrast-weighted images at different orientations are acquired by rotation of the acquisition plane for each image around one fixed encoding axis. A second warping operator $\boldsymbol{M}_{\boldsymbol{\theta}_n}$ is introduced to model the effect of unintended rigid inter-image motion. Whereas the motion parameters for $\boldsymbol{G}_n$ are known from the acquisition, the motion parameters $\{\boldsymbol{\theta}_n\}_{n=1}^N$ for $\boldsymbol{M}_{\boldsymbol{\theta}_n}$ are unknown, and have to be estimated from the data. The implementation of $\boldsymbol{G}_n$ is identical to that of $\boldsymbol{M}_{\boldsymbol{\theta}_n}$, which will now be discussed.

In what follows, for ease of notation, the LR image index n is dropped. Furthermore, the elements of a single rigid motion parameter vector $\boldsymbol{\theta}$ are indexed numerically as $\boldsymbol{\theta} = \left(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\right)$. In other words, $\theta_1, \theta_2, \theta_3$ correspond with the rigid translations, and $\theta_4, \theta_5, \theta_6$ with the Euler angles of the rigid rotations of (2). Similar to (Ramos-Llordén et al., 2017; Cordero-Grande et al., 2016), the rigid motion is expressed as a series of linear phase modulations in k-space:

$$\begin{aligned}&\boldsymbol{T}(\theta_1,\theta_2,\theta_3) = \mathcal{F}^H \boldsymbol{U}(\theta_1,\theta_2,\theta_3)\mathcal{F}\\ &\boldsymbol{R_1}(\theta_4) = \mathcal{F}_2^H \boldsymbol{V}_1^{\tan}(\theta_4)\mathcal{F}_2\mathcal{F}_3^H \boldsymbol{V}_1^{\sin}(\theta_4)\mathcal{F}_3\mathcal{F}_2^H \boldsymbol{V}_1^{\tan}(\theta_4)\mathcal{F}_2\\ &\boldsymbol{R_2}(\theta_5) = \mathcal{F}_3^H \boldsymbol{V}_2^{\tan}(\theta_5)\mathcal{F}_3\mathcal{F}_1^H \boldsymbol{V}_2^{\sin}(\theta_5)\mathcal{F}_1\mathcal{F}_3^H \boldsymbol{V}_2^{\tan}(\theta_5)\mathcal{F}_3\\ &\boldsymbol{R_3}(\theta_6) = \mathcal{F}_1^H \boldsymbol{V}_3^{\tan}(\theta_6)\mathcal{F}_1\mathcal{F}_2^H \boldsymbol{V}_3^{\sin}(\theta_6)\mathcal{F}_2\mathcal{F}_1^H \boldsymbol{V}_3^{\tan}(\theta_6)\mathcal{F}_1,\end{aligned} \tag{S22}$$

where $\mathcal{F}$ represents the 3D DFT and $\mathcal{F}_k$ corresponds with the DFT along dimension d, with $d = 1, \ldots, 3$, and where the superscript $\boldsymbol{H}$ denotes the Hermitian conjugate. Both transforms are implemented using MATLAB's built-in FFT functions. In addition, $\boldsymbol{U} \in \mathbb{R}^{N_r \times N_r}$ and $\boldsymbol{V}_d \in \mathbb{R}^{N_r \times N_r}$ are the diagonal matrices that describe, respectively, the applied translation and applied shear decomposed rotations along different axes, and whose vectors $\boldsymbol{u}$ and $\boldsymbol{v}_d$ contain the diagonal elements, which are given by:

$$\begin{aligned}&\boldsymbol{u} = e^{-i(\theta_1\boldsymbol{k}_1+\theta_2\boldsymbol{k}_2+\theta_3\boldsymbol{k}_3)}\\ &\boldsymbol{v}_1^{\tan} = e^{i\tan(\theta_4/2)\boldsymbol{k}_2\circ\boldsymbol{r}_3} \qquad \boldsymbol{v}_1^{\sin} = e^{-i\sin(\theta_4)\boldsymbol{k}_3\circ\boldsymbol{r}_2}\\ &\boldsymbol{v}_2^{\tan} = e^{i\tan(\theta_5/2)\boldsymbol{k}_3\circ\boldsymbol{r}_1} \qquad \boldsymbol{v}_2^{\sin} = e^{-i\sin(\theta_5)\boldsymbol{k}_1\circ\boldsymbol{r}_3}\\ &\boldsymbol{v}_3^{\tan} = e^{i\tan(\theta_6/2)\boldsymbol{k}_1\circ\boldsymbol{r}_2} \qquad \boldsymbol{v}_3^{\sin} = e^{-i\sin(\theta_6)\boldsymbol{k}_2\circ\boldsymbol{r}_1},\end{aligned} \tag{S23}$$

where $\boldsymbol{k}_d$ is the k-space coordinate vector of the spectral image voxels along dimension d, $\boldsymbol{r}_d$ is the spatial coordinate vector of the image voxels along dimension d, and $\circ$ denotes the Hadamard product.

With this in mind, the rigid motion operator $\boldsymbol{M}_{\boldsymbol{\theta}}$ can then be rewritten as

$$\boldsymbol{M}_{\boldsymbol{\theta}} = \boldsymbol{T}(\theta_1,\theta_2,\theta_3)\boldsymbol{R}_1(\theta_4)\boldsymbol{R}_2(\theta_5)\boldsymbol{R}_3(\theta_6). \tag{S24}$$

This helps in defining the partial derivatives of $\boldsymbol{M}_{\boldsymbol{\theta}}$:

$$\frac{\partial \boldsymbol{M}_{\boldsymbol{\theta}}}{\partial \theta_k} = \begin{cases} \frac{\partial \boldsymbol{T}(\theta_1,\theta_2,\theta_3)}{\partial \theta_k} \boldsymbol{R}_1(\theta_4)\boldsymbol{R}_2(\theta_5)\boldsymbol{R}_3(\theta_6), & 1 \le k \le 3 \\ \boldsymbol{T}(\theta_1,\theta_2,\theta_3)\boldsymbol{R}_1'(\theta_4)\boldsymbol{R}_2(\theta_5)\boldsymbol{R}_3(\theta_6), & k = 4 \\ \boldsymbol{T}(\theta_1,\theta_2,\theta_3)\boldsymbol{R}_1(\theta_4)\boldsymbol{R}_2'(\theta_5)\boldsymbol{R}_3(\theta_6), & k = 5 \\ \boldsymbol{T}(\theta_1,\theta_2,\theta_3)\boldsymbol{R}_1(\theta_4)\boldsymbol{R}_2(\theta_5)\boldsymbol{R}_3'(\theta_6), & k = 6, \end{cases} \tag{S25}$$

where

$$\frac{\partial \boldsymbol{T}(\theta_1,\theta_2,\theta_3)}{\partial \theta_k} = \mathcal{F}^H \frac{\partial \boldsymbol{U}(\theta_1,\theta_2,\theta_3)}{\partial \theta_k} \mathcal{F}, \quad 1 \le k \le 3. \tag{S26}$$

$$\begin{aligned} \boldsymbol{R}_1'(\theta_4) = {} & \mathcal{F}_2^H \boldsymbol{V}_1'^{\tan}(\theta_4)\mathcal{F}_2\mathcal{F}_3^H \boldsymbol{V}_1^{\sin}(\theta_4)\mathcal{F}_3\mathcal{F}_2^H \boldsymbol{V}_1^{\tan}(\theta_4)\mathcal{F}_2 \\ & + \mathcal{F}_2^H \boldsymbol{V}_1^{\tan}(\theta_4)\mathcal{F}_2\mathcal{F}_3^H \boldsymbol{V}_1'^{\sin}(\theta_4)\mathcal{F}_3\mathcal{F}_2^H \boldsymbol{V}_1^{\tan}(\theta_4)\mathcal{F}_2 \\ & + \mathcal{F}_2^H \boldsymbol{V}_1^{\tan}(\theta_4)\mathcal{F}_2\mathcal{F}_3^H \boldsymbol{V}_1^{\sin}(\theta_4)\mathcal{F}_3\mathcal{F}_2^H \boldsymbol{V}_1'^{\tan}(\theta_4)\mathcal{F}_2, \end{aligned} \tag{S27}$$

$$\begin{aligned} \boldsymbol{R}_2'(\theta_5) = {} & \mathcal{F}_3^H \boldsymbol{V}_2'^{\tan}(\theta_5)\mathcal{F}_3\mathcal{F}_1^H \boldsymbol{V}_2^{\sin}(\theta_5)\mathcal{F}_1\mathcal{F}_3^H \boldsymbol{V}_2^{\tan}(\theta_5)\mathcal{F}_3 \\ & + \mathcal{F}_3^H \boldsymbol{V}_2^{\tan}(\theta_5)\mathcal{F}_3\mathcal{F}_1^H \boldsymbol{V}_2'^{\sin}(\theta_5)\mathcal{F}_1\mathcal{F}_3^H \boldsymbol{V}_2^{\tan}(\theta_5)\mathcal{F}_3 \\ & + \mathcal{F}_3^H \boldsymbol{V}_2^{\tan}(\theta_5)\mathcal{F}_3\mathcal{F}_1^H \boldsymbol{V}_2^{\sin}(\theta_5)\mathcal{F}_1\mathcal{F}_3^H \boldsymbol{V}_2'^{\tan}(\theta_5)\mathcal{F}_3, \end{aligned} \tag{S28}$$

$$\begin{aligned} \boldsymbol{R}_3'(\theta_6) = {} & \mathcal{F}_1^H \boldsymbol{V}_3'^{\tan}(\theta_6)\mathcal{F}_1\mathcal{F}_2^H \boldsymbol{V}_3^{\sin}(\theta_6)\mathcal{F}_2\mathcal{F}_1^H \boldsymbol{V}_3^{\tan}(\theta_6)\mathcal{F}_1 \\ & + \mathcal{F}_1^H \boldsymbol{V}_3^{\tan}(\theta_6)\mathcal{F}_1\mathcal{F}_2^H \boldsymbol{V}_3'^{\sin}(\theta_6)\mathcal{F}_2\mathcal{F}_1^H \boldsymbol{V}_3^{\tan}(\theta_6)\mathcal{F}_1 \\ & + \mathcal{F}_1^H \boldsymbol{V}_3^{\tan}(\theta_6)\mathcal{F}_1\mathcal{F}_2^H \boldsymbol{V}_3^{\sin}(\theta_6)\mathcal{F}_2\mathcal{F}_1^H \boldsymbol{V}_3'^{\tan}(\theta_6)\mathcal{F}_1. \end{aligned} \tag{S29}$$

Finally, the derivatives of the diagonal elements of $\boldsymbol{U}$ and $\boldsymbol{V}_d$ can be summarized as

$$\begin{aligned} \frac{\partial \boldsymbol{u}(\theta_1,\theta_2,\theta_3)}{\partial \theta_d} &= -i\boldsymbol{k}_d \circ \boldsymbol{u}(\theta_1,\theta_2,\theta_3) \\ \frac{\partial \boldsymbol{v}_1^{\tan}}{\partial \theta_4} &= i\left(\tfrac{1+\tan^2(\theta_4/2)}{2}\right)\boldsymbol{k}_2 \circ \boldsymbol{r}_3 \circ \boldsymbol{v}_1^{\tan} \\ \frac{\partial \boldsymbol{v}_2^{\tan}}{\partial \theta_5} &= i\left(\tfrac{1+\tan^2(\theta_5/2)}{2}\right)\boldsymbol{k}_3 \circ \boldsymbol{r}_1 \circ \boldsymbol{v}_2^{\tan} \\ \frac{\partial \boldsymbol{v}_3^{\tan}}{\partial \theta_6} &= i\left(\tfrac{1+\tan^2(\theta_6/2)}{2}\right)\boldsymbol{k}_1 \circ \boldsymbol{r}_2 \circ \boldsymbol{v}_3^{\tan} \\ \frac{\partial \boldsymbol{v}_1^{\sin}}{\partial \theta_4} &= -i\cos(\theta_4)\boldsymbol{k}_3 \circ \boldsymbol{r}_2 \circ \boldsymbol{v}_1^{\sin} \\ \frac{\partial \boldsymbol{v}_2^{\sin}}{\partial \theta_5} &= -i\cos(\theta_5)\boldsymbol{k}_1 \circ \boldsymbol{r}_3 \circ \boldsymbol{v}_2^{\sin} \\ \frac{\partial \boldsymbol{v}_3^{\sin}}{\partial \theta_6} &= -i\cos(\theta_6)\boldsymbol{k}_2 \circ \boldsymbol{r}_1 \circ \boldsymbol{v}_3^{\sin}. \end{aligned} \tag{S30}$$

Note that this warping operator $\boldsymbol{M}_{\boldsymbol{\theta}_n}$ can be shown to be unitary (Ramos-Llordén et al., 2017), which means that its inverse is given by $\boldsymbol{M}_{\boldsymbol{\theta}_n}^H$. Hence, the motion operator $\boldsymbol{M}_{\boldsymbol{\theta}_n}$ is reversible, i.e. when applied to an image, this image can be retrieved by applying $\boldsymbol{M}_{\boldsymbol{\theta}_n}^H$ to the output of this operation.

2.2. Blurring operator

The blurring operator $\boldsymbol{B}$ in (1) describes the point spread function (PSF) of the MRI signal acquisition process. For multislice acquisition methods that sample a rectangular part of k-space, the 3D PSF is separable and can be modeled as the product of three 1D PSFs that are applied in the orthogonal directions aligned with the MR image coordinate axis. The PSFs in the frequency and phase encoding direction are defined by the rectangular part of k-space that is regularly sampled. In this work, an in-plane 2D PSF is constructed as a convolution of two identical Gaussian functions, with a standard deviation set to $0.25 \times \Delta_{\text{in-plane}}$, with $\Delta_{\text{in-plane}}$ the in-plane resolution (Van Reeth et al., 2015). The remaining through-plane 1D PSF models the slice selection profile (SSP), as SRR relies on rotated SSP cross-talk to enhance the through-plane resolution while keeping the in-plane resolution fixed. In a multislice MRI acquisition, each slice is excited by incorporating a slice selective gradient which is often generated by applying either a (windowed) sinc or a Gaussian shaped RF pulse. In this work, the SSP in the slice-direction (i.e. the z-direction) corresponds to a windowed sinc slice excitation, and was modeled as a smoothed box function (Poot et al., 2010):

$$\mathrm{SSP}(z;\Delta S) = \begin{cases} 1 & \left|\frac{z}{\Delta S}\right| \leq \frac{1}{3} \\ \frac{1}{2} - \frac{1}{2}\sin\left(3\pi\left(\left|\frac{z}{\Delta S}\right| - \frac{1}{2}\right)\right) & \frac{1}{3} < \left|\frac{z}{\Delta S}\right| < \frac{2}{3} \\ 0 & \frac{2}{3} \leq \left|\frac{z}{\Delta S}\right| \end{cases} \tag{S31}$$

where the full width at half maximum (FWHM) of the smoothed box equals the given slice thickness ΔS of the modeled LR images $\boldsymbol{s}_n$. The spatially invariant blurring of the separable 3D PSF is performed using cyclic convolution, as described in (Hansen et al., 2006), where the blurring operator $\boldsymbol{B} \in \mathbb{R}^{N_r \times N_r}$ and its conjugate transpose $\boldsymbol{B}^H \in \mathbb{R}^{N_r \times N_r}$ are spectrally decomposed as:

$$\boldsymbol{B} = \mathcal{F}_3^H \boldsymbol{\Lambda}_3 \mathcal{F}_3 \mathcal{F}_{12}^H \boldsymbol{\Lambda}_{12} \mathcal{F}_{12} \tag{S32}$$

$$\boldsymbol{B}^H = \mathcal{F}_3^H \boldsymbol{\Lambda}_3^H \mathcal{F}_3 \mathcal{F}_{12}^H \boldsymbol{\Lambda}_{12}^H \mathcal{F}_{12} \tag{S33}$$

with $\boldsymbol{\Lambda}_{12}$ the spectrum of a block-circulant-with-circulant-blocks matrix that describes the in-plane convolution, and $\boldsymbol{\Lambda}_3$ a sparse diagonal matrix whose diagonal elements are the Fourier coefficients of the first column of a circulant blurring matrix created by circularly shifting the SSP array preceeding row forward. Furthermore, $\mathcal{F}_{12}$ and $\mathcal{F}_3$ denote the 2D unitary DFT along the in-plane dimensions ($d = 1$ and $d = 2$) and the unitary 1D DFT along the through-plane dimension ($d = 3$), respectively.

2.3. Downsampling operator

Downsampling along the through-plane direction is required to resample the HR image to a LR image with increased slice thickness. To allow for noninteger resampling, interpolation is required. The choice of interpolation paradigm should allow a straightforward transpose implementation for substitution in the analytical expressions of the Jacobian and Hessian of the gradient-based SRR optimization routine. Therefore, resampling was performed using cubic convolution-based interpolation, which was first introduced in (Keys, 1981). As the original proposition of this type of interpolation is put quite general and extensive, some extra choices are required regarding its computational implementation. To promote full reproducibility of our method, these choices will now be discussed.

Cubic convolution-based interpolation (CCI) (Keys, 1981; Meijering and Unser, 2003) of uniformly sampled data implies the use of an interpolation kernel $u : \mathbb{R} \rightarrow \mathbb{R}$, which determines the weights to be assigned to the samples $f_k = f(kT)$ of an original function $f : \mathbb{R} \rightarrow \mathbb{R}$ in computing the value of the interpolant g at any arbitrary $x \in \mathbb{R}$. In what follows, for ease of notation, but without loss of generality, we will use $T = 1$. CCI may then be described as

$$g(x) = \sum_{k \in \mathbb{Z}} f_k u(x - k). \tag{S34}$$

As can readily be observed from (S34), it is required that in order for g to be an interpolant, the kernel u must satisfy that $u(0) = 1$ and $u(n) = 0$ when n is any nonzero integer. A balanced trade-off between computational cost and accuracy is provided by the family of cubic convolution kernels that consist of piecewise third-degree polynomials and are once continuously differentiable. In this work, Keys' third-order cubic convolution kernel is used (Keys, 1981), which is defined as

$$u(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1 & \text{if } 0 \le |x| \le 1, \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2 & \text{if } 1 \le |x| \le 2, \\ 0 & \text{if } 2 \le |x|. \end{cases} \tag{S35}$$

This kernel has an approximation order of $L = 3$, which implies that the resulting interpolant converges to the original function as fast as the third power of the intersample distance. It also implies that the kernel is capable of reproducing polynomials up to second degree. Outside the interval $(-2, 2)$, the interpolation kernel $u(x)$ is zero. This means that only four data samples are used to evaluate the interpolant at some new position x. In practice, the original function f can only be observed on a finite interval. For values outside this interval, boundary conditions must be chosen. In our work, values outside the image matrix are assumed to have a zero weight contribution, i.e. $f_k = 0$, indicating that only three values are used to evaluate the interpolant at the outer background edges of the generated LR image.

The HR image $\boldsymbol{r}$ can be thought of as a function

$$\boldsymbol{r} : [n] \times [m] \times [o] \to \mathbb{R}, \tag{S36}$$

where $n, m, o \in \mathbb{N}$ and $\forall k \in \mathbb{N} : [k] = \{1, \dots, k\}$. For each pair of integer coordinates, it yields a HR voxel value. Following 3D volume considerations, downsampling along the third $[o]$ through-plane dimension, i.e. the slice selection dimension of $\boldsymbol{r}$, corresponds with $[n] \times [m]$ repeated one-dimensional CCI operations. A single CCI at a non-integer position a is given by

$$x'(a) = c_1 \boldsymbol{x}(\boldsymbol{p}_1) + c_2 \boldsymbol{x}(\boldsymbol{p}_2) + c_3 \boldsymbol{x}(\boldsymbol{p}_3) + c_4 \boldsymbol{x}(\boldsymbol{p}_4), \tag{S37}$$

where $\boldsymbol{p}_1, \dots, \boldsymbol{p}_4$ are the four integer valued points surrounding a, and $c_1, \dots, c_4$ are the CCI coefficients obtained by substituting $\boldsymbol{p}_1, \dots, \boldsymbol{p}_4$ in (S35). The downsampling operator $\boldsymbol{D}$ transforms HR image $\boldsymbol{r}$ into a LR image $\boldsymbol{s} = \boldsymbol{D}\boldsymbol{r}$ of which the (i, j, k)-th voxel value is obtained by

$$(\boldsymbol{D}\boldsymbol{r})(i, j, k) = x'((i, j, k)). \tag{S38}$$

Since, by (S37), (S38) is a linear combination of voxel values of $\boldsymbol{r}$, we can interpret the action of $\boldsymbol{D}$ as a matrix vector product. Where the vectors are the $[n] \times [m]$ one-dimensional HR through-plane arrays. $\boldsymbol{D}$ can be represented by a matrix, with 4 non-zero coefficients on each row, namely the CCI coefficients of (S37) at the corresponding voxel indices separated by the inter-slice distance. The adjoint operator $\boldsymbol{D}^T$ is then simply given by the matrix with the rows of $\boldsymbol{D}$ as its columns. The rows of $\boldsymbol{D}$ or equivalently, the columns of $\boldsymbol{D}^T$ can be computed on the fly, so there is no need to store these matrices explicitly. If we denote the i-th row of $\boldsymbol{D}$, i.e. the i-th column of $\boldsymbol{D}^T$ by $\boldsymbol{o}_i$, then the action of $\boldsymbol{D}^T$ on a vector $\boldsymbol{s} \in \mathbb{R}^N$ can be implemented as follows:

$$\boldsymbol{D}^T \boldsymbol{s} = \sum_{i=1}^{N} s_i \boldsymbol{o}_i. \tag{S39}$$

With this approach we obtain an exact adjoint operator $\boldsymbol{D}^T$ that can be substituted in the analytical expressions for the Jacobian and Hessian of the gradient-based optimization routine.

2.4. Computational requirements

All algorithms were written in MATLAB and partially in C++, and run on a computer with an Intel® Core™ i7-6850K hexa-core CPU with 15MB of cache clocked at 3.60 GHz, with 32 GB of RAM. The computational complexity of the proposed SRR-joint algorithm is primarily defined by the Fast Fourier Transform (FFT)-based image warping operators $\boldsymbol{M}_{\theta_n}$ and $\boldsymbol{G}_n$, as described in section 2.1. The FFT-based implementation allows to solve the inverse SRR problem using exact adjoint image warping, and avoids inaccuracies caused by an approximate inverse of the motion. Furthermore, $\boldsymbol{M}_{\theta_n}$ is analytically differentiable w.r.t. $\boldsymbol{\theta}_n$. To speed up reconstruction, the FFT's of these image warping operators are executed on the GPU, reducing reconstruction time by a factor of 2-6 compared to pure MATLAB code, mainly dependent on the number of LR images and corresponding image dimensions. In addition, as discussed in section 1.1, MATLAB parallel computing tools were used to estimate $\boldsymbol{\theta}_n$ for each value of n separately when solving problem (P.2) of the alternating minimization method. Similarly, voxel-wise NLLS model fitting during the initialization step of the different SRR frameworks was performed in a parallel manner. The modified Bessel functions required to calculate the negative log-likelihood function with Rician PDF and the upwind TV prior term, as described in sections 2.2.3-2.2.4 of the main body, were implemented using custom C++ MEX-files for use with MATLAB. Also, to avoid excessive memory usage, the Hessian matrix of problem (P.2) was implemented using a Hessian multiply function, which gives the result of a Hessian-times-vector product without computing the Hessian directly. Bearing in mind these implementation details, and given the (rather strict) tolerance criteria described in section 3 of the main body, the reconstruction using SRR-joint took approximately 8.67 hours for a simulated LR T_1-weighted dataset, 6.43 hours for the *in vivo* T_2-weighted dataset, and 14.02 hours for the *in vivo* T_1-weighted dataset, respectively. Overall, it is expected that a more advanced implementation of the framework using only C/C++ and GPU/CUDA programming will lead to further reduction of the reconstruction time. In particular, we would like to highlight a CUDA implementation for exact adjoint image warping designed to run on NVIDIA GPUs (Renders et al., 2021), which could potentially be used to speed up the present implementation of the SRR-joint framework. Finally, this proof-of-concept implementation treats $\boldsymbol{M}_{\theta_n}$ and $\boldsymbol{G}_n$ as separate operators. However, the input of both operators could be combined to limit the number of FFT's and improve the computational efficiency.

3. Supplementary information on the whole brain simulation and *in vivo* experiments

3.1. Realistic motion parameters

For the synthetic whole brain Monte Carlo simulation experiment, different LR Rician distributed T_1-weighted magnitude datasets were simulated. Each of the N noisy LR T_1-weighted images $\{\tilde{\boldsymbol{s}}_n\}_{n=1}^N$ was affected by rigid inter-image motion parameters $\{\boldsymbol{\theta}_n\}_{n=1}^N$ with motion components $\boldsymbol{\theta}_n = \{\theta_{nk}\}_{k=1}^6$, that were chosen equal to an estimated motion set obtained from model-based SRR with the SRR-joint framework on the T_1-weighted *in vivo* dataset to guarantee realistic head movement. To indicate the order of magnitude of these rigid motion parameters, Table S1 reports the extreme and mean values for each respective motion component, calculated over the LR image index n. More specifically, the mean value of the k^{th} motion component, i.e. $\overline{\theta}_k$, was calculated as:

$$\overline{\theta}_k = \frac{1}{N}\sum_{n=1}^{N}\theta_{nk}. \tag{S40}$$

Table S1
Extreme and mean values for each of the motion parameters that were used for the synthetic whole brain simulation experiments.

	t_x [mm]	t_y [mm]	t_z [mm]	α [degree]	β [degree]	γ [degree]
extremum	0.517	2.486	2.082	2.890	-0.538	-0.836
$\overline{\theta}_k$	0.28	1.69	0.38	0.67	-0.03	-0.54

3.2. Quantitative performance measures

Table S2 and S3 show the quantitative performance measures obtained in the whole brain simulation and *in vivo* experiments, which were already reported in the paper, but this time the corresponding standard error and 95% confidence intervals are added as complementary information. In addition, Fig. S1 and Fig. S2 show, respectively, the absolute value of the relative bias maps and the relative standard deviation maps of SRR-static, SRR-reg, and SRR-joint for the whole brain simulation experiment.

Table S2
Quantitative performance measures with standard error (SE) and 95% confidence intervals (CI) for the whole brain simulations, calculated over $N_{\mathrm{MC}} = 8$ reconstruction results, for each SRR framework.

	SRR-static			SRR-reg			SRR-joint		
	value	SE	CI	value	SE	CI	value	SE	CI
Overall rel. bias									
T_1 [%]	16.840	0.030	(16.782,16.899)	10.121	0.020	(10.081,10.161)	8.698	0.016	(8.667,8.731)
ρ [%]	64.530	4.014	(56.663,72.398)	31.875	2.083	(26.793,34.957)	24.075	1.712	(20.719,27.431)
Overall rel. std. dev.									
T_1 [%]	0.316	0.001	(0.314,0.318)	0.591	0.002	(0.589,0.595)	0.686	0.002	(0.682,0.692)
ρ [%]	0.842	0.039	(0.765,0.919)	1.072	0.034	(1.006,1.139)	1.313	0.052	(1.211,1.415)
Overall rel. RMSE									
T_1 [%]	16.855	0.030	(16.796,16.914)	10.176	0.020	(10.136,10.216)	8.799	0.016	(8.767,8.831)
ρ [%]	64.547	4.014	(56.679,72.415)	30.945	2.083	(26.863,35.028)	24.204	1.713	(20.846,27.561)
RMMSE									
t_x [mm]	0.374	0	n/a	0.095	0.004	(0.085,0.105)	0.063	0.006	(0.049,0.079)
t_y [mm]	1.381	0	n/a	0.322	0.018	(0.280,0.364)	0.027	0.009	(0.006,0.049)
t_z [mm]	1.543	0	n/a	0.289	0.037	(0.203,0.375)	0.043	0.017	(0.003,0.083)
α [degree]	1.207	0	n/a	0.335	0.049	(0.219,0.453)	0.019	0.006	(0.005,0.035)
β [degree]	0.245	0	n/a	0.109	0.013	(0.079,0.139)	0.015	0.003	(0.007,0.023)
γ [degree]	0.261	0	n/a	0.079	0.004	(0.069,0.089)	0.008	0.003	(0.001,0.017)

Table S3
Quantitative performance measures with standard error (SE) and 95% confidence intervals (CI) for the *in vivo* experiment.

	SRR-static			SRR-reg			SRR-joint		
	value	SE	CI	value	SE	CI	value	SE	CI
Average edge width									
T_1 map [mm]	3.878	0.389	(2.995,4.762)	3.759	0.423	(2.801,4.718)	3.671	0.412	(2.738,4.605)
ρ map [mm]	2.920	0.357	(2.111,3.729)	2.499	0.402	(1.587,3.413)	2.389	0.468	(1.328,3.450)
$\mathrm{SNR}_{\mathrm{VOI}}$ in T_1 map									
white matter	97.103	5.094	(88.064,107.246)	99.656	5.336	(90.073,109.803)	133.730	8.161	(119.206,150.323)
CSF	30.131	7.861	(19.287,43.518)	32.412	6.393	(21.860,43.253)	34.769	8.944	(22.384,48.906)
caudate nucleus	35.211	16.113	(17.852,65.443)	38.697	14.549	(21.356,64.944)	45.074	16.729	(24.307,75.419)
$\mathrm{SNR}_{\mathrm{VOI}}$ in ρ map									
white matter	66.084	4.097	(57.727,73.382)	85.208	4.153	(77.596,93.347)	93.667	4.996	(84.195,102.947)
CSF	19.632	3.688	(14.403,25.933)	22.897	4.189	(16.213,29.394)	19.456	4.046	(14.295,26.428)
caudate nucleus	42.494	2.946	(37.241,48.165)	55.883	4.904	(47.988,65.498)	59.989	5.136	(51.068,69.679)

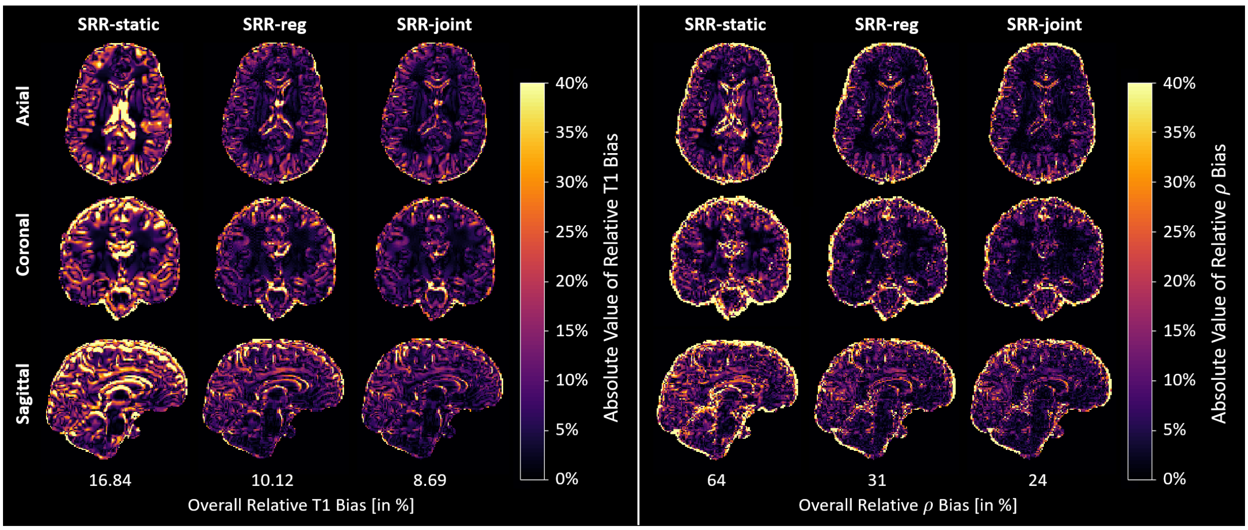


Figure S1: Absolute value of the relative bias maps for T1 and ρ, calculated from the reconstruction results of the synthetic whole brain simulations. For each of the different model-based SRR frameworks orthogonal mid-slice views are shown. Numbers at the bottom of the images indicate the overall relative bias measure, which was obtained by calculating the spatial mean of the absolute value of the corresponding relative bias map.

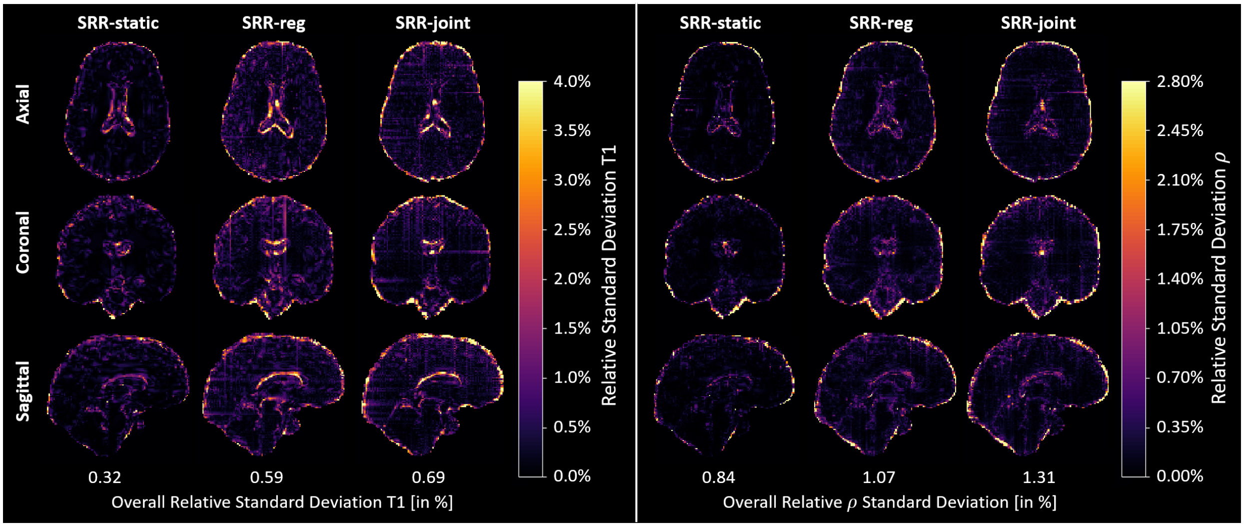


Figure S2: Relative standard deviation maps for T1 and ρ, calculated from the reconstruction results of the synthetic whole brain simulations. For each of the different model-based SRR frameworks orthogonal mid-slice views are shown. Numbers at the bottom of the images indicate the overall relative standard deviation measure, which was obtained by calculating the spatial mean of the corresponding relative standard deviation map.

3.3. Edge profile selection

As described in subsection 3.3 of the main body paper, spatial resolution of the obtained parameter maps was assessed in all 3 image dimensions by measuring the average width over 15 edge profiles. The sample of edge profiles was selected in one parameter map, and then consistently compared across all the parameter maps of the respective frameworks. To illustrate this approach, Fig. S3 shows the edge profile fitting drawn across 3 lines-of-interest for different inserts along each orthogonal plane.

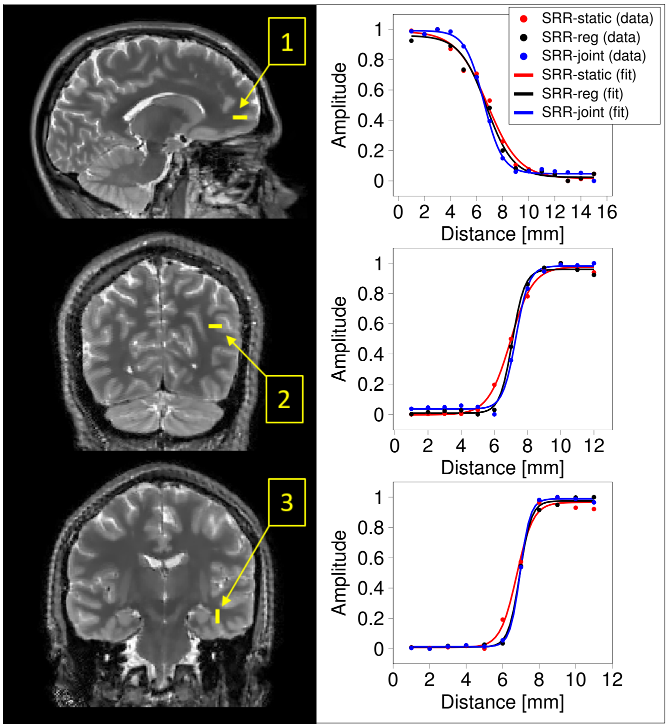
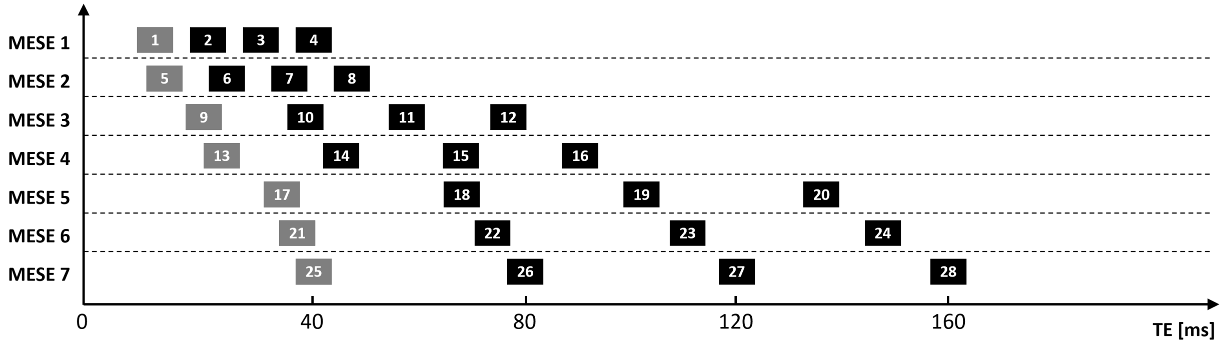
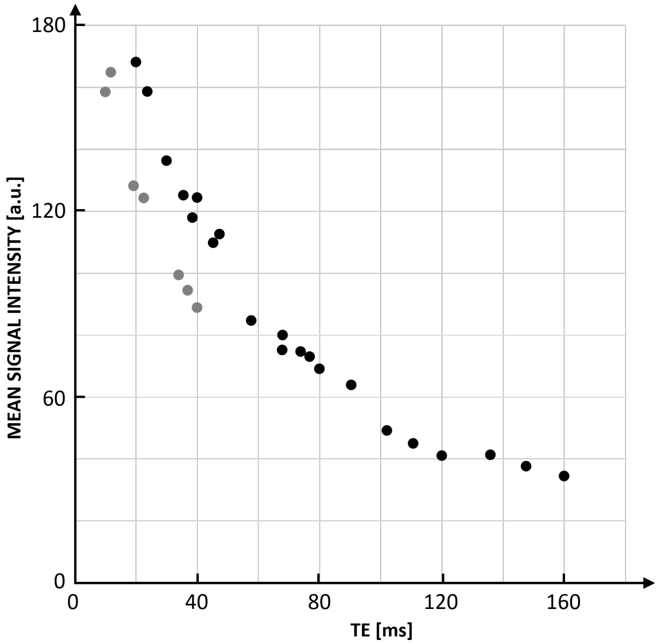


Figure S3: Spatial resolution assessment by means of edge profile fitting, drawn across 3 lines-of-interest for different inserts along each orthogonal plane, and compared for the three SRR frameworks.

3.4. Echo time selection

Figure S4 illustrates the echo time selection for the *in vivo* T_2 mapping experiment. The echo time spacing ΔTE in each MESE was chosen as such to guarantee full coverage of the T2 relaxation curve when all 7 MESE acquisitions are combined. The first echo time of each MESE was ignored in the model-based SRR to alleviate the effect of stimulated secondary echoes.





MESE NUMBER	ECHO TIMES [ms] TE$_1$	TE$_2$	TE$_3$	TE$_4$
MESE 1	1	2	3	4
	10.0	20.0	30.0	40.0
MESE 2	5	6	7	8
	11.8	23.6	35.4	47.2
MESE 3	9	10	11	12
	19.2	38.4	57.6	76.8
MESE 4	13	14	15	16
	22.6	45.2	67.8	90.4
MESE 5	17	18	19	20
	34.0	68.0	102.0	136.0
MESE 6	21	22	23	24
	36.9	73.8	110.7	147.6
MESE 7	25	26	27	28
	40.0	80.0	120.0	160.0

Figure S4: Echo time selection for the *in vivo* T_2 mapping experiment: Seven different MESE acquisitions were used to acquire a total of 28 T_2-weighted LR images. Each MESE acquisition was characterized by a unique rotation around the phase-encoding axis (rotation angles similar as in Fig. 1 in section 3.1 of the main body). Furthermore, each MESE consisted of 4 unique echo times, which are tabulated (bottom right), and visualized with the corresponding LR image number (top). The first echo time of each MESE was ignored in the model-based SRR to alleviate the effect of stimulated secondary echoes. This effect is clearly distinguishable when plotting the mean signal intensity for each LR T_2-weighted image as a function of the echo time (bottom left).

Supplementary References

Barral, J.K., Gudmundson, E., Stikov, N., Etezadi-Amoli, M., Stoica, P., Nishimura, D.G., 2010. A robust methodology for in vivo T1 mapping. Magn. Reson. Med. 64, 1057–1067. doi:10.1002/mrm.22497.

Coleman, T.F., Li, Y., 1994. On the convergence of reflective Newton methods for large-scale minimization subject to bounds. Math. Program. 67, 189–224. doi:10.1007/BF01582221.

Cordero-Grande, L., Teixeira, R.P.A.G., Hughes, E.J., Hutter, J., Price, A.N., Hajnal, J., 2016. Sensitivity encoding for aligned multishot magnetic resonance reconstruction. IEEE Trans. Comput. Imaging 2, 266–280. doi:10.1109/tci.2016.2557069.

Hansen, P.C., Nagy, J.G., O'Leary, D.P., 2006. Deblurring Images: Matrices, Spectra, and Filtering (Fundamentals of Algorithms 3) (Fundamentals of Algorithms). Society for Industrial and Applied Mathematics, USA.

Keys, R.G., 1981. Cubic convolution interpolation for digital image processing. IEEE Trans. Acoust., Speech, Signal Process , 1153–1160doi:10.1109/TASSP.1981.1163711.

Meijering, E., Unser, M., 2003. A note on cubic convolution interpolation. IEEE Trans. Image Process. 12, 477–479. doi:10.1109/TIP.2003.811493.

Poot, D.H.J., Van Meir, V., Sijbers, J., 2010. General and efficient super-resolution method for multi-slice MRI, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010, pp. 615–622. doi:10.1007/978-3-642-15705-9_75.

Ramos-Llordén, G., den Dekker, A.J., Van Steenkiste, G., Jeurissen, B., Vanhevel, F., Van Audekerke, J., Verhoye, M., Sijbers, J., 2017. A unified maximum likelihood framework for simultaneous motion and T_1 estimation in quantitative MR T_1 mapping. IEEE Trans. Med. Imag. 36, 433–446. doi:10.1109/TMI.2016.2611653.

Renders, J., Sijbers, J., De Beenhouwer, J., 2021. Adjoint image warping using multivariate splines with application to four-dimensional computed tomography. Med. Phys. 48, 6362–6374. doi:10.1002/mp.14765.

Van Reeth, E., Tan, C.H., Tham, I.W., Poh, C.L., 2015. Isotropic reconstruction of a 4-D MRI thoracic sequence using super-resolution. Magn. Reson. Med. 73, 784–793. doi:10.1002/mrm.25157.