Retrospective Motion Correction for Magnetic Resonance Imaging

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th>Citation</th>
<th>Haskell, Melissa West. 2019. Retrospective Motion Correction for Magnetic Resonance Imaging. Doctoral dissertation, Harvard University, Graduate School of Arts &amp; Sciences.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citable link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029528">http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029528</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA</a></td>
</tr>
</tbody>
</table>
Retrospective Motion Correction for Magnetic Resonance Imaging

A dissertation presented
by
Melissa West Haskell
to
The Committee on Higher Degrees in Biophysics

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Biophysics

Harvard University
Cambridge, Massachusetts
April 2019
Retrospective Motion Correction for Magnetic Resonance Imaging

Abstract

Magnetic resonance imaging, or MRI, is an important tool for diagnosing disease and studying the human body. MRI provides exceptional soft tissues contrast compared to other medical imaging methods, and is essential for the diagnosis and treatment of many diseases. However, since its inception MRI has suffered from artifacts due to patient motion. Patient motion during MRI scans causes image artifacts that degrade diagnostic utility, often requiring repeated scans, patient callbacks, or lost diagnostic potential.

Due to the prevalence and impact of motion in MRI, many techniques have been developed to detect and correct for patient motion, such as external tracking, MR motion tracking navigators, image entropy minimizations, and alternating optimizations. While all successful on some level, previous methods have sufficient weaknesses or side-effects to preclude their widespread clinical use. The goal of this work is to create a data consistency based retrospective motion correction method that can be translated to clinical use.

Here we present two complementary methods for retrospective motion correction of brain MRI images: TArgeted Motion Estimation and Reduction (TAMER) and Network Accelerated Motion Estimation and Reduction (NAMER). TAMER solves for an uncorrupted MR image by jointly optimizing for the image and the unknown rigid-body motion of a subject’s head during the scan. We minimize the data consistency error of a SENSE+motion forward model, and use reduced modelling to make the optimization computationally feasible. NAMER builds on the TAMER method by
incorporating a convolutional neural network (CNN) into the image reconstruction to accelerate algorithm convergence.

TAMER simulation results show the improved optimization search direction accuracy using a reduced model and the potential for computational speedup. NAMER simulation results show how the inclusion of the CNN in the optimization allows for a truly parallel motion optimization to be constructed, which improves algorithm convergence and accuracy. TAMER phantom imaging results show the improvement in image quality over standard reconstructions when compared to a ground truth image. In vivo imaging experiments of healthy adult subjects and a neonatal subject show the potential of using TAMER and NAMER for motion mitigation in a clinical setting.
## Contents

1 Background

1.1 Magnetic resonance imaging and the problem of patient motion .......................... 1
1.2 Previously developed motion correction techniques ............................................. 4
1.2.1 Prospective motion correction ....................................................................... 5
1.2.2 Retrospective motion correction ................................................................... 5
1.3 Model-based retrospective motion correction ...................................................... 6
1.4 Machine learning for motion correction .............................................................. 9

2 Motion forward model for MRI

2.1 Incorporation of motion into parallel imaging reconstruction model .................. 10
2.2 Joint optimization of motion trajectory and motion mitigated image .................. 11
2.3 Need for model reduction and algorithm acceleration ....................................... 14

3 Reduced Modeling for Retrospective Motion Correction ................................. 18

3.1 TAMER: TArgeted Motion Estimation and Reduction ......................................... 19
3.1.1 Model reduction using target voxels .............................................................. 19
3.1.2 TAMER algorithm ....................................................................................... 21
3.1.3 Target voxel selection - correlation matrix method ....................................... 26
3.2 TAMER experiments ....................................................................................... 30
3.2.1 Simulations ................................................................................................. 30
3.2.2 Phantom experiments .................................................................................. 32
3.2.3 In vivo imaging ............................................................................................ 33
3.3 TAMER results ............................................................................................... 33
3.3.1 Simulations ................................................................................................. 34
3.3.2 Phantom experiments .................................................................................. 38
3.3.3 In vivo imaging ............................................................................................ 40

4 Accelerated Motion Correction Using Machine Learning .............................. 42

4.1 NAMER: Network Accelerated Motion Estimation and Reduction .................. 43
4.1.1 MRI acquisition ......................................................................................... 43
4.1.2 Training data generation ............................................................................ 45
4.1.3 Motion artifact detecting convolutional neural network .......................... 46
4.1.4 Motion parameter optimization ........................................................... 47
4.1.5 Model-based image reconstruction using CNN-guided motion estimate .... 50

4.2 NAMER experiments ............................................................................. 51
  4.2.1 Simulations ......................................................................................... 51
  4.2.2 In vivo experiments ........................................................................... 52

4.3 NAMER results ...................................................................................... 52
  4.3.1 Simulations ......................................................................................... 53
  4.3.2 In vivo imaging .................................................................................. 56

5 Future Directions and Conclusion .......................................................... 58
  5.1 Summary of Work .................................................................................. 58
    5.1.1 Image space model reduction using the TAMER method ............... 59
    5.1.2 Accelerated motion correction using the NAMER method ............. 61
  5.2 Future Directions ................................................................................. 62
    5.2.1 Combination of TAMER and NAMER approaches ....................... 62
    5.2.2 Improvements for 3D motion correction ........................................ 63
    5.2.3 Motion forward model improvements .......................................... 64
    5.2.4 Implementation improvements ...................................................... 64
    5.2.5 Convolutional neural network optimization .................................. 65
  5.3 Conclusion ............................................................................................ 65

References ................................................................................................... 75
Listing of figures

1.1 Effect of subject motion on MRI images .................................................. 2
1.2 MRI motion sensitivity .............................................................................. 3
1.3 Common motion artifact mitigation strategies ......................................... 4
1.4 Model-based image reconstruction .......................................................... 7

2.1 Illustration of the Forward Model ............................................................ 12
2.2 Goal of retrospective motion correction .................................................. 15

3.1 TAMER forward model decomposition ................................................... 20
3.2 TAMER Overview ...................................................................................... 22
3.3 Coil grouping ............................................................................................. 23
3.4 Target Voxel Selection .............................................................................. 28
3.5 TAMER results in data with simulated motion ......................................... 35
3.6 Motion Search Direction Accuracy .......................................................... 36
3.7 Comparison of Target Voxel Selection Methods ..................................... 37
3.8 Image Results for Target Voxel Selection Methods .................................. 38
3.9 TAMER Translation Phantom Results ...................................................... 39
3.10 TAMER Rotation Phantom Results ........................................................ 39
3.11 In Vivo Results ......................................................................................... 40
3.12 In Vivo TAMER Results Compared to State-of-the-Art ......................... 41

4.1 NAMER method overview ................................................................. 44
4.2 Convolutional neural network for motion artifact detection .................... 47
4.3 How to determine if intra-shot correction is needed ................................. 50
4.4 CNN artifact mitigation across whole brain ............................................. 53
4.5 NAMER compared to alternating method in simulated motion data ........ 54
4.6 Simulated intra-shot motion correction .................................................. 55
4.7 NAMER in vivo inter- and intra-shot motion correction ......................... 56
4.8 Full volume NAMER motion correction ................................................ 57
This thesis is dedicated to my grandmothers, Elizabeth Cahill Haskell and Dorothy Irene Wuest LePage.
None of this work would have been possible without the support of many, many people, and I want to thank them with my sincerest gratitude.

First, I want to thank my thesis advisor Larry Wald, and my co-advisor Stephen Cauley. Larry, I feel so lucky to have had the opportunity to work in your lab. Your scientific creativity and pursuit of new ideas is something that I deeply admire, and getting to work with you on this project has made for an incredibly rewarding Ph.D. experience. Steve, when I started this project I had basically no idea what I was doing, and I am incredibly grateful for all of your help and everything you taught me over the years. Your insights and knowledge helped to propel this project forward when it needed it the most.

I want to thank my dissertation advisory and thesis committee members, Ellen Grant, Elfar Adalsteinsson, Bruce Rosen, and Susie Huang, for all they have done for me. Ellen, thank you for serving on my dissertation advisory committee, for providing me with a clinical perspective, and for connecting me with other groups working on motion correction. Elfar, taking your course on MRI gave me the foundational knowledge for this project and got me excited about the field of MRI. You are one of the kindest faculty members I have ever met, and your support and input on my research over the years have been invaluable. Bruce, you believed in me as a researcher before I believed in myself, and your encouragement has meant the world to me. I deeply appreciate the commitment you have to students at the Martinos Center. Susie, thank you to you and David for letting me help out in developing the HST164 class and for providing a clinical perspective on my research. I admire you immensely and am so grateful to have you as a role model.

Thank you so much to Jim & Michele for everything you do for the Biophysics Program. I am so grateful for how you always supported me, especially in my early years when I didn’t feel like I belonged at Harvard.

Thank you to all my fellow MR-PIG group members. Coming to work everyday is pretty great when I get to be surrounded by such an amazing group of people. I also want to thank the rest of my friends at the Martinos Center, especially my fellow Ph.D. students, for adding lots of fun and laughter to my time here. I want to especially thank Erica Mason, I don’t know how we would have gotten through graduate school without each other and I’m so grateful for your friendship. Thank you also to all my friends outside of graduate school who keep my life fun and interesting beyond the scope of my academic endeavors.

I am beyond fortunate to have such an amazing nuclear and extended family, and am also so
lucky to have gained such an incredible family-in-law. My family has always been the constant throughout my life, and I deeply appreciate the support I have received from them. Thank you especially to my mom, Maggie, my dad, R.D., and my sister, Jen. Growing up as a nerdy kid I often didn’t feel like I fit in, but your unconditional love and support over the years allowed me to pursue my academic goals and find a field that I love.

Lastly, thank you to my husband, Dan. Your love helped me get through the lowest of lows in these last six years, and all of the achievements of which I’m most proud would not have been possible without your support. There are no words to describe the happiness I feel to have you as my partner in life.
1.1 Magnetic resonance imaging and the problem of patient motion

Magnetic resonance imaging, or MRI, is an important tool for diagnosing disease and studying the human body. MRI provides exceptional soft tissues contrast compared to other medical imaging methods, and is essential for the diagnosis and treatment of many diseases. MRI is also an incredibly flexible technique with many different contrast mechanisms available for diagnosis of neurological
Figure 1.1: Effect of subject motion on MRI images. A. A sagittal slice of a $T_1$-weighted image with and without subject motion. B. An axial slice of a $T_2$-weighted image with and without subject motion.

However, since its inception MRI has suffered from artifacts due to patient motion (see Fig. 1.1). Subject movement is common in MRI because of the long scan times required, and also because many patients experience anxiety during an MRI scan, making it understandably more challenging for them to remain still. The resulting motion artifacts can lead to obscured anatomical features such as vascular structures or small lesions. Studies have also shown that head motion during MRI acquisitions can significantly affect the analysis of brain pathology. This leads to the frequent use of sedation or anesthesia during MRI scans, which increases patient risk, can increase the scan time by $>2x$, and can increased the cost up to 9x compared to non-sedated imaging. A 2015 study at a US hospital showed that 20% of all MRI scans (and 29% for ED and inpatient exams) needed to be repeated, leading to an estimated cost of $115,000/scanner/year. In the worst case scenario, a radiologist will simply “read-through” artifacts, missing possible pathologies that go untreated.

Subject motion is a problem for many medical imaging modalities, but it is particularly relevant in MRI because of the longer scan times and the Fourier based acquisition. Individual MRI scans are on the order of minutes and a full MRI exam with multiple scans can be 30 minutes to an hour (which is much longer compared to other modalities, such as X-ray computed tomography (CT) which is on the order of seconds). Ideally a subject would be still for each of the multi-minute MRI
scans, and even if they could hold still for a single scan, by the end of a 30+ minute exam, patient discomfort can make that even more challenging. Remaining still for a long period of time is also especially challenging for certain subjects, including pediatric, geriatric, and psychiatric patients.

In addition to the longer scan time, MRI raw data in acquired in the Fourier domain, commonly referred to in the MRI field as “k-space”. The object being images is transformed to k-space by applying linearly varying magnetic fields in multiple dimensions, creating a sinusoidal pattern across the image that represent different spatial frequencies. By acquiring many different spatial frequencies, one can “fill” the k-space data matrix of a certain object, and then perform image reconstruction based on the inverse Fourier transform to create an image from the raw k-space data. Motion artifacts arise because the raw data acquired in k-space is inherently sensitive to subject movement as shown in Fig. 1.2. Slight differences in the position of the anatomy being imaged can lead to significant phase distortions in k-space. This results in ringing or blurring artifacts when the k-space data is
<table>
<thead>
<tr>
<th>Motion prevention</th>
<th>Artifact reduction</th>
<th>Motion correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>Faster imaging</td>
<td>Navigators</td>
</tr>
<tr>
<td>Distraction</td>
<td>Insensitive sequences</td>
<td>Self-navigated trajectories</td>
</tr>
<tr>
<td>Feed and wrap (for babies)</td>
<td>Gradient moment nulling</td>
<td>Prospective correction</td>
</tr>
<tr>
<td>Foam restraints</td>
<td>Saturation bands</td>
<td>Retrospective correction</td>
</tr>
<tr>
<td>Sedation</td>
<td>Triggering and gating</td>
<td></td>
</tr>
<tr>
<td>Bitebars/head holders</td>
<td>Phase reordering</td>
<td></td>
</tr>
<tr>
<td>Breathhold</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.3: Common motion artifact mitigation strategies.** This table from Zaitsev, et al. 2015⁶ outlines the various categories and types of motion mitigation strategies.

reconstructed into image space⁶.

1.2 PREVIOUSLY DEVELOPED MOTION CORRECTION TECHNIQUES

The prevalence and negative impact of motion artifacts has led to the development of many different motion mitigation and correction strategies⁷,⁸,⁹ (see Fig. 1.3). The simplest way to avoid artifacts is to prevent patient motion from occurring, which can be done using subject training, foam restraint, or sedation. Artifacts can also be reduced by shortening the scan time or using less motion sensitive sequences.

When motion cannot be prevented or mitigated using simple strategies, motion correction is performed. Note that in this work “motion correction” is used to describe the correction of motion artifacts within a single time point scan, as opposed to registration, which is the alignment of different time points to each other. Motion correction techniques can be categorized broadly as either prospective (subsection 1.2.1) or retrospective (subsection 1.2.2). In prospective methods, the imaging coordinate system defined by the MR scanner is updated based upon estimates of the patient movement throughout the scan, whereas in retrospective methods the imaging coordinate system is constant throughout the acquisition and the data is corrected after the scan.
1.2.1 Prospective motion correction

Prospective techniques require repeatedly measuring the patient motion during the acquisition to update the scanner coordinates. This can be accomplished using external detectors, image space navigators, and FID navigators. However, all of these methods add complexity to the exam. External markers require placement and calibration, and navigators can be disruptive to the acquisition sequence timing. If the scanner coordinate system is changed on the fly, the motion sensor must also be highly accurate to avoid introducing erroneous motion into the acquisition. Together, these problems have limited the use of prospective motion correction in the clinic.

1.2.2 Retrospective motion correction

Retrospective techniques correct motion artifacts after the k-space data have been acquired. This lessens the concern of imperfect motion information corrupting the acquisition, since standard reconstructions can always be chosen if they give better results. Retrospective techniques include motion information into a physical model to describe the effect of patient motion on the k-space data, for example using a matrix formulation. In the case of brain imaging, it is often assumed that head motion can be described using a rigid-body model, where the six rigid-body motion parameters (three for translation, three for rotation) are included in the image reconstruction. A translation in image (i.e., object) space results in a phase ramp in k-space, and a rotation in image space results in a rotation in k-space. By incorporating the effect of the translations and rotations into the forward model, these methods account for motion and reconstruct a motion artifact free image.

One approach to finding the motion parameters for a retrospective correction is to measure them directly. Studies have used motion information both from MR navigators or from external tracking devices. Periodic motion models for cardiac and respiratory imaging have also been applied to retrospective motion mitigation in body imaging by binning data into different motion
Determining motion directly from the k-space data offers an alternative to using measured motion information from navigators or tracking devices. This approach is advantageous because it requires no modifications to the MRI exam procedure. In this case, only the raw k-space data is used to search for the motion parameters by either minimizing an image quality metric associated with motion, or by jointly estimating the motion parameters and image based on data consistency (i.e. a model-based approach, see next for more detail). In autofocusing\cite{16,17}, the image entropy or a similar criteria was minimized. Autofocusing techniques have been implemented for rigid\cite{18}, multirigid\cite{19}, and nonrigid\cite{20,21} motion correction. Joint optimization of the image and motion has also been explored in the context of retrospective motion correction in PET/CT\cite{22} and PET/MR\cite{23,24,25,26,27}.

### 1.3 Model-based Retrospective Motion Correction

The motion correction methods used in this work build off the general concept of model-based image reconstruction\cite{28}. In model-based image reconstruction, the acquired signal (k-space data), $s$, is modeled as a multiplication of the encoding matrix, $E$, and the unknown image volume, $x$:

$$ s = E x $$ \hspace{1cm} (1.1)

The encoding matrix, $E$, is also referred to as the “forward model” that maps from the image domain to the k-space signal domain. This is represented pictorially in Fig. 1.4.

To find the unknown image, Eqn. 1.1 must be inverted. For a small $E$, its inverse, $E^{-1}$, could be computed, and then $x$ could be found simply by multiplying the inverse encoding model by the acquired signal:

$$ x = E^{-1} s $$ \hspace{1cm} (1.2)
Figure 1.4: Model-based image reconstruction. A. A 2D or 3D image is considered as a 1D vector containing all of the image voxels. B. The unknown image, $\mathbf{x}$, is operated on by the encoding matrix, $\mathbf{E}$, (also known as the forward model operator) to generate the measured signal ($k$-space data), $\mathbf{s}$. This formulation sets up the image reconstruction problem as a matrix inversion problem.

However, in practice this is unfeasible for MRI image reconstruction because of the large size of $\mathbf{E}$. Instead, an estimate of the image, $\hat{\mathbf{x}}$, can be found by minimizing the difference between the acquired data and the model (i.e., the data consistency error), using the conjugate gradient method

$$
\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \| \mathbf{s} - \mathbf{E} \mathbf{x} \|_2
$$

(1.3)

In the simplest MRI situation, the encoding matrix, $\mathbf{E}$, is the discrete Fourier transform matrix and represents taking the Fourier transform of the image to generate k-space data. With the extension of MRI to multiple receive coils, the encoding model also expanded to include the sensitivity maps for multiple coil elements. This extension has allowed the development of parallel imaging, where k-space data can be undersampled to drastically reduce image acquisition time. Model-based image reconstruction can also include the effects of field inhomogeneities.
nonlinear fields\textsuperscript{48,49}, gradient imperfections\textsuperscript{50}, or non Cartesian sampling strategies such as radial\textsuperscript{51,52,53}, spiral \textsuperscript{54}, or wave-CAIPI\textsuperscript{55,56}. Equation 1.3 can also be extended to include regularization terms, such as sparsity terms in compressed sensing approaches\textsuperscript{57} or low rank constraints\textsuperscript{48,49,60}.

In this work and others\textsuperscript{61,62,63,64}, model-based reconstruction is used to retrospectively correct for motion by including subject motion in the encoding matrix\textsuperscript{57,61,62,63}. The final image is found by searching for the motion parameters and corresponding image that minimize the data consistency error of the motion forward model. The details on how to implement this mathematically are described in Chapter 2.

This approach utilizes the intrinsic encoding of motion within multi-channel array coils\textsuperscript{65}. Motion information is extracted from the k-space data by estimating both the rigid-body position parameters present for each shot of the image acquisition and the image itself via the forward model inversion. This has been previously done using an alternating minimization approach, either with an initial estimate of the motion from external sensors as in the GRICS method\textsuperscript{64}, or in a completely retrospective fashion\textsuperscript{62,63}. In our work, we introduce a reduced image reconstruction model to improve the joint-optimization procedure. In the TArgeted Motion Estimation and Reduction (TAMER) method (Chapter 3), we address the computational challenge of the joint image-motion optimization by limiting intermediate motion search steps to targeted subsets of the image, which have been carefully chosen based on model separability. This approach is similar to reduced model approaches used for fast auto-calibration of gradient trajectories for wave-CAIPI MRI reconstruction\textsuperscript{65}. In the Network Accelerated Motion Estimation and Reduction (NAMER) method (Chapter 4), we extend upon the TAMER method by incorporating a convolutional neural network into the image reconstruction. This improves algorithm separability and convergence, making a data consistency approach more feasible.
1.4 Machine learning for motion correction

In addition to using model-based methods, this thesis also examines the use of machine learning (ML) techniques for retrospective motion correction. ML methods provide a potential avenue for dramatically reducing the computation time and improving the convergence of retrospective motion correction methods. Recent work has demonstrated how ML can be used to detect, localize, and quantify motion artifacts and deep networks have been trained to reduce motion artifacts. While the reliance on an ML approach alone shows promise, issues remain with the degree of artifact removal and the robustness of the process to the introduction of blurring or non-physical features. In Chapter 4, we attempt to harness the power of a convolutional neural network (CNN) within a controlled model-based reconstruction. In Chapter 4 we demonstrate how ML can be effectively incorporated into retrospective motion correction approaches based on data consistency error minimization. Other works have balanced data consistency error with ML generated image priors (created using variational networks) to dramatically reduce reconstruction times and improve image quality for highly accelerated acquisitions.
 Motion forward model for MRI

In this chapter we describe the forward model used to mathematically represent the signal acquired during an MRI exam with patient motion. The model is based on the well-known SENSitivity Encoding (SENSE) reconstruction algorithm. In the original SENSE implementation, the main goal was to use sensitivity maps from multiple receive coils to reconstruct undersampled data (i.e., data below the Nyquist sampling rate). Here, we will use the extra degrees of freedom afforded by a multicoil acquisition to include the effects of motion into the MRI acquisition.
2.1 Incorporation of motion into parallel imaging reconstruction model

The SENSE\textsuperscript{43} model can be described as:

\[ s = UFCx \]  \hspace{1cm} (2.1)

where \( C \) contains the spatially varying coil sensitivities, \( F \) is the Fourier encoding operator, and \( U \) is the undersampling operator. We can group \( C, F, \) and \( U \) into a signal encoding model \( E \), as in Eqn. 1.1.

To represent a SENSE-based rigid-body motion forward model, we can extended Eqn. 1.1 as follows to describe the signal acquired in a 2D multishot imaging sequence:

\[ s = E_\theta x \]  \hspace{1cm} (2.2)

\[ E_\theta = \begin{pmatrix}
E_{\theta_1} \\
E_{\theta_2} \\
\vdots \\
E_{\theta_{N_{sh}}}
\end{pmatrix} \]  \hspace{1cm} (2.3)

where \( x \) is a \( N \times 1 \) column vector of the \( N \) image voxel values, \( E_\theta \) is the \( NC \times N \) forward model operator (encoding matrix) for a given \( M \times 1 \) patient motion trajectory \( \theta \), and \( s \) is the \( NC \times 1 \) multicoil signal data from \( C \) coils. \( E_\theta \) is the concatenation of the encoding model for each of the \( N_{sh} \) shots (\( M = 6N_{sh} \) for the six rigid-body motion parameters at each shot). The encoding model for each individual shot, \( l \), can be described as:

\[ E_{\theta_l} = U_l F C T_{xy,l} T_{z,l} R_l \]  \hspace{1cm} (2.4)
Figure 2.1: Illustration of the Forward Model. The k-space data for a given shot \( l \) is related to the 3D image volume to be reconstructed, \( x \), through the following operators: (1) image motion, which includes rotations, \( R_l \), through-plane translation, \( T_{z,l} \), and in-plane translations, \( T_{xy,l} \) (2) weighting by the coil sensitivity matrix \( C \) (3) Fourier encoding operator \( F \) (4) k-space data for the individual shots are created using the undersampling operator \( U_l \).

where for shot \( l \), \( R_l \) is the rotation operator, \( T_{z,l} \) is the through-plane translation operator, \( T_{xy,l} \) is the in-plane translation operator, \( C \) contains the spatially varying coil sensitivities, \( F \) is the Fourier encoding operator, and \( U_l \) is the \( nC \times NC \) undersampling operator, where \( n \) is the number of k-space samples acquired per shot \( (N_{sh}n = N) \). See Fig. 2.1 for an illustration of the motion forward model.

The motion vector, \( \theta \), is a concatenation of the head positions at the time of each shot as parameterized by the six the rigid body motion parameters. Thus, \( \theta \) is of dimension \((6N_{sh} \times 1)\) and is the time-series for all six motion variables. The motion parameters for a given shot, \( l \), are the three translation parameters and the three rotation parameters. The matrix \( R_l \) performs a 3D rotation the image volume, \( x \), by applying three sequential rotations about each axis, based on the motion that occurred at shot, \( l \), in the following order: (1) roll, \( \theta_{3N_{sh}+l} \) degrees about I-S, (2) yaw, \( \theta_{4N_{sh}+l} \) degrees about A-P, and (3) pitch, \( \theta_{5N_{sh}+l} \) degrees about R-L. We used the MATLAB (Mathworks, Natick,
MA, USA) built-in bilinear interpolation rotate function for all rotations, and our model allows for an offset to the center of rotation. The matrix \( T_{z,l} \) shifts the image volume in the through-plane direction by \( \theta_{2N_{sh}+l} \) voxels using sinc interpolation between the slices. This sinc interpolation can be expressed as

\[
T_{z,l} = F_z^{-1} T'_{z,l} F_z
\]

where \( F_z \) is a 1D fast Fourier transform (FFT), \( T'_{z,l} \) is the \( k \)-space through-plane translation operator which applies the proper phase ramps, and \( F_z^{-1} \) is the 1D inverse FFT. Because of the reduced modeling in TAMER, these 1D inverse FFTs are only applied across certain columns of the image volume.

The matrix \( T_{xy,l} \) shifts the slices in-plane by \( \theta_{l} \) voxels in the phase encode direction, and by \( \theta_{N_{sh}+l} \) voxels in the readout direction, also using sinc interpolation

\[
T_{xy,l} = F_{xy}^{-1} T'_{xy,l} F_{xy}\]

where \( F_{xy} \) and \( F_{xy}^{-1} \) are the 2D FFT and inverse 2D FFT, respectively, and \( T'_{xy,l} \) is the \( k \)-space in-plane translation operator which applies the proper phase ramps. As the translation parameters are expressed in voxels within the vector \( \theta \), each direction must be scaled by the resolution to get the patient motion trajectory in physical parameters.

Our implementation here is for 2D contiguous slice imaging. We separate the translations operators into the through-plane and in-plane directions. The stack of contiguous 2D slices is treated as a 3D image in that through plane translations and out-of-plane rotations can move voxels from one slice into another. Thus, our general forward model assumes that all voxels can be used to match data from a given shot. In practice we break the operators into in-plane and through-plane operators, and for some of the validations, only in-plane operators were utilized. There is, of course, a difference in how an instantaneous translation (or through-plane rotation) effects 2D slice data compared to 3D Fourier encoded data. Currently our method is implemented only for the 2D slice case,
but could easily be extended to model true 3D encoding. We also note that spin-history effects were not modeled and would require a more detailed book-keeping dependent on the acquisition used.

2.2 Joint Optimization of Motion Trajectory and Motion Mitigated Image

The forward model detailed in Section 2.1 shows how subject motion can be incorporated into the image encoding model, but the ultimate goal of this work is to reconstruct motion mitigated MRI images. This is accomplished by jointly optimizing the motion trajectory \( \theta \) and the image volume \( x \) by minimizing the data consistency error:

\[
[\hat{\theta}, \hat{x}] = \arg\min_{\theta, x} \| s - E_{\theta}x \|_2
\]  

(2.5)

Accurately and efficiently performing the above minimization is the overarching goal of this thesis. By including motion in the optimization, we aim to produce high-quality images even in the presence of patient motion (see Fig. 2.2). We note that the motion parameters themselves are nuisance parameters. However, solving for the motion parameters explicitly, as opposed to only performing image-based motion artifact mitigation, allows us to verify the accuracy of our motion estimation methods (for example, in simulations and controlled phantom experiments performed in Chapters 3 & 4).

2.3 Need for Model Reduction and Algorithm Acceleration

In a typical forward model based image reconstruction, a least-squares objective function \( \| s - E_{\theta}x \|_2 \) would be minimized to estimate \( x \), e.g., using the conjugate gradient method. Typically, the unknowns would only appear in \( x \), whereas in the retrospective motion optimization problem (Eqn. 2.5), additional motion unknowns appear in \( E_{\theta} \). This corresponds to a more difficult non-
Figure 2.2: Goal of retrospective motion correction. A. When there is no motion and standard reconstruction approaches are used, a high-quality image is produced. B. When there is patient motion and standard reconstruction approaches are used, there will be motion artifacts. C. The goal of retrospective motion correction is to produce a high-quality image even in the presence of subject motion by including motion in the image reconstruction process.
linear estimation that requires repeated evaluation of linear least-squares objectives based upon motion trajectories of interest. In the case of MRI motion correction, this is especially challenging due to the number of unknowns.

For example, a standard clinical 2D $T_2^*$-weighted RARE/FSE/TSE sequence has a $0.5 \times 0.5 \times 4$ mm$^3$ voxel size, an in-plane FOV of $220 \times 220$ mm$^2$, and 35 slices, leading to an image matrix size of $448 \times 448 \times 35$, which is $>7$ million unknown voxels that need to be reconstructed. This type of clinical scan also has R=2 undersampling and 80% resolution, leading to 176 lines/slice. With an echo train length (ETL, also known as the turbo factor, TF) of 11, that implies 16 shots/slice, or 560 shots total. For rigid body motion with six motion parameters for each shot, that means there are $>3,000$ motion parameters that need to be estimated. With this many of nonlinear parameters to estimate, standard nonlinear solvers cannot be directly applied, since they are only designed for problems with a small number of unknowns (e.g., less than 10). To ameliorate the computational challenge of this joint optimization, in this thesis we introduce a model reduction scheme (Chapter 3) and algorithmic speedups using machine learning (Chapter 4).

Solving this large computational problem has also been explored in prior work using an alternating optimization approach. In that work, the algorithm alternates between performing a full-volume reconstruction of the image and then taking a single (or very small number) of optimization steps to update the motion estimates. A strength of this approach is that it breaks a difficult joint optimization problem up into two smaller ones. The benefit of this division depends on the separability of the two types of optimization variables (voxel values and motion parameters). For example, the first step to search for the motion variables occurs after a full solve of all the voxel unknowns, and is thus likely to pursue an accurate initial search direction that will improve the objective function. For subsequent iterative steps estimating motion parameters, the quality of the search directions ($\nabla$ in Eqn. 3.8) might degrade if tight coupling exists between the two types of optimization variables. In this case subsequent motion parameter estimates are not as accurate which will not
allow for the objective to be improved. At this point the alternating algorithm will re-estimate the voxel values using the most recent motion estimate. If the alternating algorithm can only take a few steps before switching to the other variable type, the efficiency is reduced. The reduced efficiency is primarily expected to impact computational speed, yet we note that the variable separation approach as implemented in 61 performs well and quickly. Their approach also makes effective use of reduced spatial resolution image data to speed up both searches and an analytic gradient calculation for the motion search direction.

Because there are uncertainties about the coupling between the voxel and motion parameters and open questions on the effect of using low spatial resolution versions of the data during the optimization, we sought an alternative approach that does not rely on complete uncoupling of the variables called TArgeted Motion Estimation and Reduction (TAMER). In Chapter 3, we employ a reduced model of the image reconstruction which can be repeated quickly for every estimation of the motion variables. In this scenario, the trade-off between accuracy and computation time is based on the number of target voxels used when searching for the motion. The data consistency of the reduced data set is evaluated for each potential motion step, capturing the effect of coupling between the variables. A hypothesis of the approach is that the search directions for the motion parameters are improved by this regular update of the chosen subset of voxel intensities. After the motion estimate has converged, a full volume solve (estimation of all the voxels) is performed to generate the final image. It is important to note that neither method (alternating nor TAMER) however can guarantee convergence to a global minimum, due to the properties of the non-linear problem being solved. It is also important to note that hybrid methods which use both alternating minimization and reduced data models could be formed.
Reduced Modeling for Retrospective Motion Correction

In this chapter we describe the inherent voxel coupling properties that occur during an MRI image acquisition and present a method, TArgeted Motion Estimation and Reduction (TAMER), that leverages this coupling to create an efficient retrospective motion correction method called. The TAMER method is described, and experiments and results are presented demonstrating its potential
as a method for clinical motion correction of MRI images.

3.1 TAMER: TArgeted Motion Estimation and Reduction

3.1.1 Model reduction using target voxels

The vector of image voxel values, $x$, is broken into two parts to decrease the size of the reconstruction at any given iteration. Specifically, the vector is broken into: $x_t$, a small targeted subset of voxels (typically 5%), and $x_f$, a fixed subset of voxels that is temporarily set constant to values associated with the best previous motion estimate iteration. By fixing a large portion of the voxels to constant values, the number of parameters to be minimized in Eqn. 2.5 can be reduced by an order of magnitude. The indices of $x_t$ are determined based on the structure of the encoding matrix, $E_\theta$.

See Section 3.1.3 for a detailed discussion of target voxel selection.

This voxel separation allows for the linear encoding model to be broken down into two encoding submatrices $E_\theta(t)$ and $E_\theta(f)$ which operate on the target and fixed voxels, respectively:

$$E_\theta x = \begin{pmatrix} E_\theta(t) & E_\theta(f) \end{pmatrix} \begin{pmatrix} x_t \\ x_f \end{pmatrix} \quad (3.1)$$

This separation of $x$ and $E_\theta$ can be done for any choice of target and fixed voxels. We can now separate the signal, $s$, into two terms: $s_t$, the signal contribution from the target voxels, and $s_f$, the signal contribution from the fixed voxels.

$$E_\theta x = E_\theta(t)x_t + E_\theta(f)x_f = s_t + s_f \quad (3.2)$$

Both $s_t$ and $s_f$ are the same length as $s$, $NC \times 1$, but they contain signal from the different voxel sets. A visual representation of Eqn. 3.2 can be seen in Figure 3.1. To isolate the signal contribution
Figure 3.1: TAMER forward model decomposition. The forward model, $s = Ex$, is broken down into components from the targeted voxels and the fixed voxels. The targeted components are updated every iteration, and the fixed components (which are computationally more burdensome to calculate) are only updated occasionally. Their sum represents the measured k-space data, $s$. This breakdown allows for TAMER to iterative only over "targeted voxels".
of the target voxels, $s_t$, we can subtract the contribution of the fixed voxels from the total signal $s$:

$$s_t = s - E_{\theta}(f)x_f = E_{\theta}(t)x_t$$  \hspace{1cm} (3.3)

The optimization can now be reduced to fitting the signal contribution of the target voxels $s_t$ to the correct target voxel values $x_t$ and the correct motion $\theta$. The non-linear joint optimization of the motion and target voxels is now the following:

$$[\hat{\theta}, \hat{x}_t] = \arg\min_{\theta, x_t} \| s_t - E_{\theta(t)}x_t \|_2$$  \hspace{1cm} (3.4)

Here, $E_{\theta}(f)x_f$ (a large, slow operation) is evaluated only once for each motion estimate $\theta$. The repeated evaluations of $E_{\theta}(t)x_t$ necessary to find $x_t$ using conjugate gradient are fast, due to the small size of $x_t$ and the ability to cache many of the operators illustrated in Fig. 2.1.

3.1.2 TAMER ALGORITHM

The TAMER method is comprised of three stages: (1) an initial SENSE reconstruction across the complete volume (assuming all motion parameters are zero), followed by the target voxel selection and channel grouping, (2) jumpstart of the motion parameter search to improve convergence of the optimization, and (3) efficient search for motion parameters using the reduced model joint optimization. A schematic overview of the method is described in Fig. 3.2.

1. Initialization: Image Estimate, Select Target Voxels, and Group Channels

First, we perform an initial reconstruction of the full image volume, $x_0$, assuming all motion parameters are zero. Here we calculate $x_0$ by minimizing the least squared error of the forward model with zero motion by solving the system

$$(E_0^HE_0)x_0 = (E_0^Hs)$$  \hspace{1cm} (3.5)$$
Figure 3.2: TAMER Overview. (1) Raw k-space data is used to reconstruct a motion corrupted image, a target voxel pattern is selected to be jointly optimized along with the motion parameters, and the coils on which to run TAMER are grouped based on observed motion artifacts (2) Jumpstart of the motion parameters through a coarse grid search, and (3) Joint optimization of motion parameters and spatially varying target voxels.
Figure 3.3: Coil grouping. (1) First, the model error assuming no motion occurred is computed for each coil. Depending on the location of the coil and the type of motion, different model inconsistencies will be present (e.g., for a patient lying supine and nodding their head "no", the pivot point is in the back of the head and there are more motion artifacts in an anterior coil). (2) The correlation of the model error for different coils is computed. Each coil is perfectly correlated to themselves (bright diagonal), and off diagonal entries with high values correspond to different coils that have highly correlated error. (3) Group matching finds the coils that are highly correlated, and TAMER is run on those subsets.

using conjugate gradient, where $E_0 \triangleq UFC$, and $E_0^H$ denotes the conjugate transpose of $E_0$. The sensitivity maps are found using the ESPIRiT method provided as part of the Berkeley Advanced Reconstruction Toolbox. Note that the sensitivity maps in the TAMER forward model are considered to be spatially fixed and independent of any coil loading differences that arise through patient movement.

Next, the indices of the initial target voxels (i.e. the voxel coupling pattern) are selected using the locality of interaction properties contained in the encoding matrix. This can be done without a priori knowledge of the k-space data or underlying image. They are determined solely by the sequence parameters and coil sensitivities (see Section 3.1.3 for a detailed discussion of target voxel selection).

Prior to jointly estimating the motion and voxels based on data consistency, the channels are
grouped by their motion artifact properties. This grouping is formed by comparing the data consistency error between the measured k-space data and the forward model with no motion, $F^{-1}\{s - E_0 x_0\}$. The 2D correlation coefficient of this metric is computed across all channels, and coils with similar motion artifacts will have high correlation (see 3.3). During the iterations to find the motion and image estimates, only the coil group with the largest correlated artifacts is used. After convergence the final motion time course is applied to the data from these coils, and a final image is reconstructed. The correlation across channels was investigated for all data sets used here, and this process can be automated by employing standard clustering algorithms.

Algorithm 1 TAMER algorithm

1. Initialization: find $x_0$, select initial target voxels, group channels
2. Jumpstart motion search
3. while $i < i_{max}$ & $\Delta \epsilon > \Delta \epsilon_{min}$ do
   3a. $s_{t(i)} = s - E_{\theta(i)}(f)x_f$
   3b. Solve for $x_{t(i)}$ using PCG:
       $$(E_{\theta(i)}^H(t)E_{\theta(i)}(t))x_{t(i)} = E_{\theta(i)}^H(t)s_{t(i)}$$
   3c. $\epsilon(i) = \|s - E_{\theta(i)}x_{t(i)}\|_2$
   3d. Calculate gradient $\nabla(i)$
   3e. Update motion: $\theta^{(i+1)} = \theta^{(i)} + \nabla(i)$
   3f. Shift target voxels
end

2. Jumpstart of Motion Parameter Search

To provide an initial guess of the patient motion for the joint optimization, the data consistency metric is evaluated over a coarse range of values for each motion parameter independently. These values can be quickly calculated in parallel, and the best value for each motion parameter is used to construct the initial search direction for the joint optimization. In our experience, generic optimization packages such as that used in this work can struggle to get out of a trivial initial search position due to lack of tuning of the optimization parameters. Here we used rotation angles within $\pm 5^\circ$, with a step size of $r$ and the translation ranges were chosen similarly but in units of voxel shifts (not de-
3. Joint Optimization of Motion Trajectory and Imaging Volume Using a Reduced Model Search

After determining the initial target voxels, motion estimate, and coil groupings, the reduced model joint optimization in Eqn. 3.4 is performed using a quasi-Newton search. Specifically, we use the standard MATLAB unconstrained non-linear optimization tool, fminunc, for ease of implementation, but algorithm performance could be improved with a custom descent algorithm such as the one used in prior work. In the TAMER method, we limit our search to only consider an optimal subset of voxel values for any given motion estimate. Thus, the motion trajectory, \( \theta \), contains the \( M \) optimization variables, and when evaluating the data consistency of any given motion estimate, only target voxels are reconstructed to find an estimate of the joint data consistency error.

For a given motion trajectory estimate at search step \( i \) (i.e. \( \theta^{(i)} \)), the reconstructed target voxels, \( x_t^{(i)} \), are found using conjugate gradient by solving the system:

\[
(E^H_{\theta^{(i)}}(t)E_{\theta^{(i)}}(t))x_t^{(i)} = E^H_{\theta^{(i)}}(t)s_t^{(i)}
\]  

(3.6)

\( s_t^{(i)} \) is found using Eqn. 3.3, where the initial values of \( x_f \) are the values of \( x_0 \) at the corresponding indices. Once \( x_t^{(i)} \) is calculated, the estimated data consistency error for step \( i \) can be evaluated using the full forward model:

\[
e^{(i)} = \| s - E_{\theta^{(i)}}x^{(i)} \|_2
\]  

(3.7)

This represents one evaluation of the objective function for input motion trajectory \( \theta^{(i)} \).

To calculate the gradient at step \( i \) across all \( M \) motion parameters in \( \theta^{(i)} \), we find the change in data consistency error for a small change in \( \theta^{(i)} \), keeping \( x^{(i)} \) constant.

\[
\nabla^{(i)} = e^{(i)} = \| s - E_{\theta^{(i)}} + s_\theta x^{(i)} \|_2
\]  

(3.8)
The new motion estimate is updated using this finite difference gradient approximation

\[
\theta^{(i+1)} = \theta^{(i)} + \nabla^{(i)} 
\]  

(3.9)

As the joint optimization progresses, the target voxel pattern is shifted across the image. Voxel values at the previous target indices are updated in \( x_f \). The search continues until the maximum number of steps has been taken, or the data consistency converges. It is possible that the high frequency portions of k-space are not resolved in the joint optimization, due to the dominance of the low k-space error in the data consistency minimization. To overcome this potential problem, the joint optimization in Eqn. 3.4 can be modified such that the high k-space errors are weighted more heavily in the minimization (such as with an inverse Gaussian filter). This reduced model search allows for a fast evaluation of many motion trajectories, while also ensuring the motion estimates have high data fidelity by reconstructing the target image volume. TAMER can also be parallelized in several ways. The conjugate gradient call (step 3b) can be parallelized to solve for \( x_f^{(i)} \) at each step, or different target voxel sets can be optimized in parallel. The gradient evaluation (step 3d) can be computed in parallel as well.

3.1.3 Target voxel selection - correlation matrix method

The reduced model optimization in Eqn. 3.4 is most effective when voxels that are coupled strongly to each other are optimized together. Voxel coupling in MRI occurs whenever there is ambiguity in the encoding of the MRI signal, e.g. undersampled and/or motion corrupted Cartesian acquisitions and many non-Cartesian acquisitions. For example, in a motion free Cartesian \( R = 2 \) SENSE image reconstruction, voxels that are located FOV/2 apart in the phase encode direction are coupled to each other.

Mathematically, voxel coupling is described using the correlation matrix, \( E^H E \). The correlation
matrix is very large and is typically never explicitly calculated in full. However, the coupling seen by an individual voxel can be easily computed by applying the forward operator, \( E \), and then the adjoint operator, \( E^H \), to a delta function, i.e. \( (E^H E)\delta_j = (E^H E)_j \). The magnitude of the entries in this column of the correlation matrix represent the strength of interaction of each voxel on the voxel \( j \), (i.e. \( x_j \)). We will refer to this voxel, \( x_j \), as the “root voxel” which will be used to form the target voxel pattern. For a fully sampled, motion-free Cartesian acquisition, each voxel is only coupled to itself, and the correlation matrix would be diagonal.

The first column of Fig. 3.4 shows an example of a subset of \( E^H E \) for no motion, measured patient motion, and random motion, all for RARE \(^{82}\) imaging with \( R = 2 \) undersampling and echo train length (ETL) of 8, which represents the number of k-space lines collected per shot (ETL is also referred to as the turbo factor). For the no motion case, only two voxels in each column are non-zero, because only two voxels are coupled as a result of the \( R = 2 \) undersampling in k-space. However, when the encoding matrix includes patient movement, the correlation matrix shows an increased level of voxel coupling. When random motion is used instead of a patient’s measured motion, the observed coupling pattern is similar. Thus, we utilize random motion as a proxy to the unknown motion trajectory in order to easily determine the most likely coupling pattern for the voxels.

The second column of Fig. 3.4 shows a single column of the correlation matrix, \( (E^H E)_j \), for no motion, patient motion, and random motion. A single column of the correlation matrix represents the magnitude of coupling between a root voxel and each of the other imaging voxels. For this example, the root voxel is shown with a black “X” on the image in Fig. 3.4. These values are plotted in image space in the third column of Fig. 3.4. As one would expect for the two motion cases, the spacing between coupled voxels in image space is the FOV/ETL. This is because for RARE imaging, each shot can be thought of as an individual motion free image that is highly undersampled. The aliasing pattern from this type of undersampling will alias voxels spaced by FOV/ETL in the phase.
Figure 3.4: Target Voxel Selection. First Column: Correlation matrix $E_{\theta}^H E_{\theta}$ for the voxels shown in the white outlined ROI on the left (log scale). Second Column: For a given root voxel (i.e. one row of the correlation matrix), the correlation of the other voxels to the root voxel is plotted (linear scale). Third Column: The magnitude of each voxel’s correlation to the root voxel is plotted in image space (log scale). Fourth Column: The target voxels are chosen for a given region of correlated voxels using thresholding. The motion values used in the second row are from the motion trajectory measured during an fMRI scan of a patient with Alzheimer’s disease.
encode direction, leading to the coupling properties seen in Fig. 3.4.

To create a specific target voxel pattern, a “root” voxel is chosen, and the column of the correlation matrix with the same index as the root voxel is computed. For all the examples presented in this paper, the center image voxel was used as the “root” voxel. The voxel coupling values are then thresholded and smoothed to create the target voxel set (fourth column of Fig. 3.4). This target voxel pattern can then be used in the reduced model optimization.

The target voxels are changed throughout the optimization by shifting the initial target voxels perpendicular to the phase encode (and therefore voxel coupling) direction. This follows a similar procedure to that of domain decomposition for solving linear systems of equations. Specifically, in order to converge to the global solution for a fully coupled problem, the boundary conditions for any subset of voxels needs to be correct. Thus, we shift the target voxel pattern across the imaging region during each step of the optimization. We shift the target voxels perpendicular to the phase encode direction by roughly the diameter of the disks shown in Fig. 3.4. Once the pattern has done a full “sweep” across the FOV, we shift the pattern along the readout direction and repeat. To create 3D target voxel subsets, we consider the target voxel regions to be uniform across all slices. Thus, the circular target voxel subsets shown for a single slice example in Fig. 3.4 are cylinders extending across all slices for a multi-slice imaging volume.

Note that the target voxels for a true motion trajectory and a random motion trajectory are very similar, hence we do not need to know the patient motion a priori to create an accurate target voxel subset. This technique is applicable to many other acquisition types when bulk rigid motion occurs, but the method’s performance for non-rigid body motion has not been investigated in this work, although the target voxel approach might be useful in non-rigid motion models.
3.2 TAMER experiments

The ability of the TAMER optimization to mitigate motion artifacts was tested using simulations, phantom data, and human subject data. Simulations were performed to verify TAMER’s ability to correct for motion, to compare its search accuracy to an alternating implementation, and to compare different target voxel subsets. Validation also included acquiring corrupted data from a translating anthropomorphic head phantom, a rotating pineapple phantom, and *in vivo* brain data from a healthy adult subject. To compare to the work in\(^6\), we have reconstructed a subset of their provided neonate motion-corrupted data, and compared the image results.

3.2.1 Simulations

1. TAMER Method Validation

   The validation data were acquired on a 3T Siemens Trio scanner (Siemens Healthcare, Erlangen, Germany) with the standard Siemens 32-channel head array coil. T\(_2\)-weighted RARE images were acquired with 22.4 x 22.4 mm\(^2\) field of view, resolution 0.5 x 0.5 x 3.0 mm\(^3\), TR = 6.1 s, TE = 98 ms, flip angle 150°, and R = 1. For the simulation experiments a single slice from the data was down-sampled to a 128 x 128 matrix size (1.75 x 1.75 mm\(^2\) resolution) and used as the ground truth k-space data.

   A modified version of the forward model in Eqn. 2.4 was applied to the motion-free dataset to generate a dataset with motion effects where the gold-standard motion-free image was also known (note the term for the coil sensitivities (\(C\)) was omitted since we apply the motion-corrupting forward model to individual coil data). The simulated motion time-course was from an Alzheimer’s disease patient’s motion trajectory found using motion alignment of functional MRI timeseries data\(^8\). We selected a segment of the full motion trajectory that included both in-plane rotation and in-plane translation, but no through-plane motion, to investigate motion effects on a single slice.
The 2D version of the TAMER algorithm was then applied to the slice of motion corrupted k-space data assuming standard protocol parameters, \( R = 2 \) undersampling, and turbo factor = 16. Reconstructed images were compared to the ground truth no motion images to verify the algorithm’s ability to converge to the ground truth motion.

The TAMER reduced model allows for motion transformations to be performed across small regions of the 3D FOV, and restricts the regions that FFTs need to be applied and would allow for specific cached DFT matrices to be used. These properties hold as we continuously shift the target voxel pattern across the volume during TAMER. To investigate the computational cost of the target voxel updates vs. full volume reconstruction, we performed repeated evaluations of the objective function for the reduced and full model using the 128 x 128 image, but without the use of cached local DFTs.

2. **Targeted vs. Alternating Motion Parameter Search Accuracy**

We demonstrate using simulations the potential advantages of a reduced model joint minimization over an alternating minimization. We implemented an alternating minimization where at each step we calculated: (1) the search direction of an alternating method with no voxel updates (2) the search direction when updating a small subset of target voxels, and (3) the full model search direction that would be determined after performing a full-volume solve (which can be thought of as the ground truth search direction). The alternating motion gradient’s and the targeted motion gradient’s directionalities were then compared to that of the full model.

3. **Comparison of Voxel Selection Methods**

To investigate TAMER’s dependence on voxel coupling properties, the convergence for various target voxel subsets were investigated. Three different voxel subsets were compared to the correlation matrix method: (1) a rotated correlation matrix (i.e. voxels along the readout direction instead of the phase encode direction are used), (2) uniformly spaced voxel disks (of the same size as the disks used in the correlation matrix method), and (3) randomly spaced voxel disks. The convergence
curves of each method were compared, as well as their respective output images. Note that these voxel selection methods were used here for comparison only, and all proceeding experiments use the correlation matrix method for voxel selection as described in Section 3.1.3.

3.2.2 Phantom experiments

To test the ability of TAMER to mitigate translational movement, an anthropomorphic head phantom was placed on a translation stage and moved intermittently throughout the scan in the anterior-posterior direction. Data were acquired on a 3T Siemens Trio scanner with a 12-channel birdcage array coil using a standard Siemens 2D T2-weighted RARE sequence. Sequence parameters include TR = 3.0 s, TE = 99 ms, ETL = 11, 220 x 220 mm² FOV, 12 slices, resolution 0.9 x 0.9 x 3 mm³, and R = 1. The target voxels were 3.4% of the total voxels depending on the slice. Each slice of the volume was reconstructed independently using the 2D version of TAMER since the motion was restricted to in-plane.

TAMER’s ability to suppress rotation artifacts was examined by reconstructing phantom data from a pineapple that was rotated in-plane throughout the scan using a motion actuator. Data were acquired on a 1.5 T Siemens Avanto scanner with a 12-channel head matrix coil using a standard Siemens 2D T2-weighted RARE sequence. Sequence parameter were ETL = 16, 230 x 208 mm² FOV, 0.6 x 0.6 mm² in-plane resolution, 5 mm slice thickness, TR = 3.8 s, TE = 93 ms, refocus angle = 150°, and R = 1. The coil data was automatically compressed to eight channels by the vendor hardware. The target voxels were 2.7% of the total voxels. 2D TAMER was first performed on this dataset as described in equation 3.4. The objective function was then minimized again using the first result as a starting point, but with an inverse Gaussian filter weighting high frequency k-space components. This seems to have been helpful in resolving fine structures in the image. Given the noise promotion of this reweighting, we choose to denoise the final image using Tikhonov regularization, although this step may be excluded if desired, and does not affect the motion estimates (i.e.,
Tikhonov regularization was not used in the minimization objective function).

3.2.3 In vivo imaging

The TAMER optimization was tested on a human adult subject who was asked to shake their head during the middle of the scan to create motion artifacts. The 2D version of TAMER motion correction was applied here for in-plane motion only on a single slice. The instructions given to the subject were to “nod no” and the axial slice prescription was chosen to create mainly in-plane motion effects; however, unlike the phantom experiments, the experiment is not fully controlled, and it is possible that through-plane motion might also have occurred. The data were acquired using ETL = 11, 220 x 220 mm² FOV, 0.9 x 0.9 mm² resolution, 3 mm slice thickness, TR = 3.0 s, TE = 99 ms, refocus angle = 150°, and R = 1 on a 3T Siemens Trio scanner with a 32-channel head array coil. The target voxels were 2.4% of the total voxels.

To validate the image quality of our method compared to the state-of-the-art, we reconstructed the data provided with prior work from https://github.com/mriphysics/multiSliceAlignedSENSE/releases/tag/1.0.1. The data set provided was for a neonatal subject with uncontrolled motion, and the dataset was down sampled in the slice direction to remove the slice oversampling. Five slices were then reconstructed using TAMER 3D motion correction and the prior method, performing both in-plane and through-plane motion correction on the R=2.49 undersampled data. To provide similar image results, the settings for the prior method’s reconstruction were also altered to turn off outlier rejection, slice smoothing, and Gibbs filtering.

3.3 TAMER results

We present results showing the ability of the TAMER optimization to mitigate motion artifacts. Simulation results show the algorithm’s accuracy when a ground truth motion trajectory is known.
In addition, simulations are used to compare the search direction of a targeted vs. alternating approach, and to compare the convergence between various voxel selection methods. Experimental results include reconstructing motion corrupted data from a translating anthropomorphic head phantom, a rotating pineapple phantom, and in vivo brain data from both a healthy adult subject and a neonatal dataset provided in the accompany code of prior work.63

3.3.1 Simulations

1. TAMER Method Validation

Figure 3.5 shows the results of applying the TAMER method to simulated motion data. This k-space data was corrupted by both translations and rotations, resulting in an image RMSE of 20.2% compared to a ground truth image. The image improves when jointly optimizing the image and translational motion parameters, resulting in an 8.4% image RMSE. The image is further improved by correcting both translation and rotation motion parameters, resulting in a 3.5% image RMSE.

Using this simulated example again (128 x 128 image matrix size, single slice), we measured an overall 17x computational speedup when using target voxel updates compared to the full volume solves, with a speedup for each of the individual steps of: motion operators 82x, coil sensitivities 4x, Fourier transform 5x, and the sampling operator 4.5x.

2. Targeted vs. Alternating Search Accuracy

Figure 3.6 shows the motion search direction accuracy of the motion parameters relative to the “gold standard” search direction. The motion search direction (\( \nabla \)) in Eqn. 3.8 is the difference between the motion estimates at each step of the search (i.e., \( \theta^{(i+1)} = \theta^{(i)} + \nabla^{(i)} \)). The “gold standard” is taken as the search direction found after a full volume solve of the image voxels. In this assessment, a full volume solve of all the voxels is done, followed by 13 iterations of the motion parameter search. This process is then repeated 30 times. If no update of the voxel information occurs during the motion iterations, this would represent an alternating optimization with 13 motion i-
Figure 3.5: TAMER results in data with simulated motion. (Top) Motion corrupted image. Right column shows difference image formed from the gold standard (uncorrupted) image. (Middle) Image reconstructed using TAMER when only correcting for translations. (Bottom) Image reconstructed when optimizing for both in-plane translation and rotation. Percentages in bottom right correspond to image space RMSE compared to ground truth.
Figure 3.6: Motion Search Direction Accuracy. Here we compare the accuracy of the motion search direction (\( \nabla \)) in Eqn. 3.8, which is the difference between the motion estimates at each step of the search, i.e., \( \theta^{(i+1)} = \theta^{(i)} + \nabla^{(i)} \), when updating a subset of target voxels, versus assuming the image is constant (no voxel updates). A. The motion search direction error is plotted as a function of the alternating step (i.e. how many times the image volume has been reconstructed). B. The motion search direction is plotted for the no voxel updates and targeted voxel updates method. The first row shows the search direction at step 2 of the motion optimization, and the second row shows the search direction at step 11 of the motion optimization. Each method (solid) is compared to the full model search direction (dotted), and the average % difference is shown in the top right of each plot.

operations per spatial solve. By construction, the error is 0\% for the first of the 13 motion iterations, but grows as the motion search progresses in the absence of updated voxel information, so we plot the average error of the 13 iterations. Fig. 3.6-A (solid gray curve) shows that the average search error over the 13 motion steps is relatively constant at about 73\% for the 30 alternations when no voxel information is updated. In the dotted curve in Fig. 3.6-A, the first motion search direction is computed the same way; based on the full volume solve of the voxel data. But the remaining 12 motion estimate iterations are informed by an evaluation of the target voxels. In this case, the average search direction error is lowered by the continuous re-evaluation of the target voxels. Fig. 3.6-B displays the actual motion search direction across the nine motion parameters (for this in-plane simulation with three shots, we had three rotations parameters and six translation parameters for translation in each of the two in-plane directions, i.e. \( M = 9 \)). Early in the search (motion optimization step 2), both methods find the proper search direction. However after 11 of the 13 steps, the alternating approach was highly inaccurate (95\% error), whereas the targeted approach had 14\% error.
3. Comparison of Voxel Selection Methods

Figure 3.7 shows the convergence of the TAMER optimization for various target voxel selection methods. The data consistency error is plotted as a function of optimization step (i.e. number of motion trajectories tested). The initial data consistency error of the motion-free ground truth SENSE reconstruction used in the simulations was 4.15% (some amount of model error is expected due to coil sensitivity inaccuracies along with image noise). With simulated motion corruption, the data consistency error increased to 12.3%, showing the additional disagreement between the multi-coil data and the forward model when assuming no patient motion. The correlation matrix method converged most quickly, followed by the rotated correlation matrix, then random voxel locations, and finally uniformly spaced locations.

Figure 3.8 shows each method’s image once it has reached a data consistency error below 6%, with the number of steps required shown in the bottom right. To achieve this degree of motion artifact mitigation, 88 optimization steps were required with the correlation matrix method, and 135,
Figure 3.8: Image Results for Target Voxel Selection Methods. (Left) Simulated motion corrupted image. (Right) TAMER images for each target voxel selection method. The white number in the bottom right of each image shows the number of steps taken to arrive at the image (all corresponding to 6% data consistency RMSE).

142, and 158 steps were required for the rotated correlation matrix, random spacing, and uniform spacing, respectively.

3.3.2 Phantom experiments

Results from the translation experiment can be seen in Fig. 3.9. Two representative slices are shown where TAMER mitigates the effects of motion on the final image. Ringing artifacts are removed, and the image space RMSE for the two slices decreases from 20.5% and 15.4% to 15.1% and 10.7%, respectively. The output motion trajectories found using TAMER only show displacement in the A-P direction (axis of translation stage movement), with no significant motion detected in the R-L direction, in agreement with the orientation of the translation stage. Similar motion trajectories are seen for the two slices, consistent with rigid-body motion. For the top slice in Fig. 3.9, TAMER took 50 minutes running on a Dual Intel Xeon E5-2690 v4 CPU. Source code and data for this figure has been uploaded to https://github.com/mwhaskell/tamer_mri.

Figure 3.10 shows rotation corrupted and TAMER corrected images for a rotating pineapple. Blurring has been reduced using TAMER, and fine detailed structures that were indiscernible in the motion corrupted image are resolved, although signs of blurring remain. Image RMSE compared
Figure 3.9: TAMER Translation Phantom Results. (Left) Comparison of corrupted and motion corrected anthropomorphic phantom images from RARE acquisition with resolution 0.9x0.9x3 mm$^3$ (error scaled by 7x). Motion was created using a 1D physical translation stage along the A-P direction. (Right) Estimated motion trajectories found using TAMER.

Figure 3.10: TAMER Rotation Phantom Results. A pineapple was rotated in-plane throughout the scan using a motion actuator, with estimated magnitudes of rotation up to ±3°. After TAMER correction, the high frequency components of the object were restored that were previously not visible due to rotation corruption. RMSE compared to a no motion ground truth was calculated for each image.
to a no motion ground truth was 17.4% using standard (motion corrupted) reconstruction, and 10.0% using TAMER. The data consistency RMSE was initially 27.6% for the motion corrupted scan, improved to 25.0% following the first TAMER pass, and decreased to 24.5% after the second (i.e. weighted) TAMER pass (for comparison, a no motion acquisition of the same phantom with the same settings had a data consistency RMSE error of 23.4%).

3.3.3 In vivo imaging

Figure 3.11 shows the results of motion corrupted and TAMER corrected in vivo data. Fig. 3.11-A shows the correlation coefficients of the image space error of each individual coil. High error correlation was present in the anterior coils, while other channels contained few motion artifacts and were considered to measure mostly non-motion corrupted data. Fig. 3.11-B shows ringing was reduced by TAMER, while maintaining the high-resolution content of the image. The data consistency error dropped from 16.8% in the original motion corrupted reconstruction to 14.6% using the TAMER method, demonstrating the motion model's ability to better fit the acquired data.
The clustering produced nine coils that were highly correlated with large data-consistency errors, which were uncorrelated to the remaining channels that showed low data consistency error. The rigid-body model employed in this work will not accurately describe data that has these attributes and we think it may be a result of confounding effects, e.g., spin history. By restricting motion correction to this group of channels we were able to achieve better image quality improvement than an optimization across all channels. For the other experiments, we did not observe disjointed groups when the coil artifacts were analyzed, and all the channels were placed in a single group.

In order to validate the accuracy of our approach to the state-of-the-art\textsuperscript{63}, we modified the alternating based optimization package provided as part of that work to closely coincide with the optimization objective shown in Eqn. 2.5. Fig. 3.12 show the image results when using an alternating method similar to the one found in Cordero-Grande, et al. 2018\textsuperscript{61} and when using TAMER. The TAMER data consistency error decreased from 14.5% to 13.1%. The code provided with Cordero-Grande, et al. 2018\textsuperscript{61} used image space error as a stopping criteria (described in Cordero-Grande, et al. 2016\textsuperscript{62} App. A), and therefore the data consistency error was not provided as an output of the algorithm. The TAMER method is able to produce comparable image quality to that of the alternating method.
Accelerated Motion Correction Using Machine Learning

In this chapter, we show how a convolutional neural network (CNN) trained to remove motion artifacts from images can improve a model-based motion estimation. During each pass of the iterative process, a CNN image estimate is used as an image prior for the motion parameter search. This motion estimate is then used to create an improved image which can be propagated as input to the
CNN to initialize the next pass of the algorithm. The quality of this image estimate can significantly improve the conditioning and convergence of the nonconvex motion parameter optimization. In addition, we demonstrate that with this high quality CNN image estimate, the motion optimization becomes separable. The separability of the motion model allows for small sub-problems to be optimized in a highly parallel fashion. This allows for a scalable extension of the model-based approach to include intra-shot (line-by-line) motion effects as well as the inter-shot motion effects in the RARE acquisition. The increased computation speed also facilitates implementation on standard vendor computation hardware.

4.1 NAMER: Network Accelerated Motion Estimation and Reduction

An overview of the Network Accelerated Motion Estimation and Reduction (NAMER) method is shown in Fig. 4.1. The method is initialized with a SENSE reconstruction of the raw k-space data. The motion correction algorithm is divided into three processes that are iteratively performed: (1) an artifact detecting CNN (Fig. 4.2) is applied to identify motion artifacts that are subsequently removed, (2) based upon the CNN output, a non-linear optimization estimates the associated motion parameters that minimize the data-consistency error of the forward model, and (3) a model-based reconstruction is performed to generate an updated image based on the motion parameters found in step (2). These steps are then repeated using the updated model-based reconstruction to further reduce the data consistency error and related motion artifacts. Example NAMER code can be found at https://github.com/mwhaskell/namer_mri.

4.1.1 MRI acquisition

sentence about how for CNN need to stick with one acquisition for everything so trained network will work on data?
Figure 4.1: NAMER method overview. First, a motion corrupted image is reconstructed from the multicoil data, assuming no motion occurred. Next, motion mitigation is performed by looping through three steps: (1) remove motion artifacts in image space by passing the 2-channel complex image (1 channel for the real component and one channel for the imaginary component) through the motion detecting CNN, (2) search for the motion parameters by minimizing the multicoil data consistency error of a motion-inclusive forward model, using complex voxel values from the CNN image, and (3) reconstruct the full image volume using the motion-inclusive multicoil forward model and position coordinates from step (2).
T₂-weighted RARE/TSE/FSE\textsuperscript{32} data were acquired on 3T MAGNETOM Skyra and Prisma scanners (Siemens Healthcare, Erlangen, Germany) using the product 32-channel head array coil and default clinical protocols. The imaging parameters are: TR/TE=6.1s/103ms, in-plane FOV=220x220mm², 4mm slices, 448x448x35 matrix size, 80% phase resolution, R=2 uniform undersampling, and echo train length (ETL)=11. A 12s FLASH\textsuperscript{85} scan provides motion robust auto-calibration data that is used to generate coil sensitivity maps (calculated using ESPRiT\textsuperscript{45} from the BART toolbox\textsuperscript{79}). In this work data from six healthy subjects were acquired in compliance with institutional practice. Data from four of the subjects were utilized to train the CNN, and the data from the two other subjects were used to evaluate the performance of our method through simulations and supervised motion experiments.

4.1.2 Training data generation

Training data for the CNN were created by manipulating raw k-space data (free of motion contamination) using the forward model described in Chapter 3 to simulate the effects of realistic patient motion trajectories. Motion trajectories were created using augmentation (shifting and scaling) of timeseries registration information from fMRI scans of patients with Alzheimer’s disease. The residual learning CNN attempts to identify the motion artifacts using an $\ell_2$-norm loss function against the motion corrupted input image minus the ground truth image. Ten evenly spaced slices from the four healthy subjects were used, and each slice was corrupted by 10 different motion trajectories. Thus, there were 400 motion examples available to choose from in order to create the training data.

The training data was refined through the exclusion of motion corrupted images with RMSE (compared to ground truth) that was greater than 0.50 or less than 0.12. Images with RMSE greater than 0.50 were excluded because they contained severe motion corruption artifacts that could bias the training (due to the large error). Images with RMSE less than 0.12 were excluded because they contained so few motion artifacts which were not productive toward training the CNN to detect ar-
tifacts. Overall 76 of the 400 examples were excluded, 61 for RMSE < 0.12 and 15 for RMSE > 0.50, leaving 324 remaining. To reduce the memory and computational footprint, 24 random cases were dropped to limit the training size to 300 motion cases. These 300 images were divided into patches of size 51x51 with a stride size of 10 (resulting in an overlap region of 41 voxels), which produced 1600 patches per motion corrupted image. For each image, 1250 out of 1600 patches were randomly selected to bring the total number of motion corrupted patches to 375k. From this set of patches, 300k were used for training and 75k used for validation (80/20 training/validation split).

4.1.3 Motion artifact detecting convolutional neural network

Figure 4.2 shows the 27 layer network topology, which follows previous work\(^3\) and is implemented in Keras.\(^{86}\) The network isolates the image artifacts within the two-channel (real and imaginary) motion corrupted input patch. The initial layer is a convolutional layer followed by a ReLU activation. The next 25 layers consist of a convolutional layer with batch normalization and ReLU activation, and the final layer is a convolution layer. The number of hidden layers was chosen because the loss function did not improve using more than 25 layers. Additional layers were not added to avoid potential problems with overfitting. Each convolutional layer uses 3x3 kernels with 64 filters. The network was trained using the Adam optimizer\(^{87}\), with learning rate = 1e-4 and a mean squared error loss function.

Before being passed through the CNN, the images are scaled to range in magnitude from 0 to 1. Patches are then created, with size 51x51 (as in the training data) and a stride size of 8, which generated 2500 patches per image. After patches were passed through the CNN, they were combined and normalized by the number of patches overlapping at each pixel. No image padding was used for the examples in this paper since the FOV was not tight on the anatomy. The updated artifact free image,
Figure 4.2: Convolutional neural network for motion artifact detection. A motion corrupted image is input as two channels (corresponding to the real and imaginary components) to a 27-layer patch-based CNN consisting of convolutional layers, batch normalization, and ReLU nonlinearities. The network outputs the image artifacts, which can be subtracted from the input image to arrive at a motion mitigated image.

\( x_{\text{cnn}} \), can be described mathematically as:

\[
x_{\text{cnn}} = x - \text{CNN}(x)
\]  

(4.1)

where \( \text{CNN}(x) \) is an image containing the detected motion artifacts within the input image \( x \).

4.1.4 Motion Parameter Optimization

some kind of connection to chapter 3?

The vector containing the motion parameters, \( \theta \), is estimated from \( x_{\text{cnn}} \) through a non-linear optimization to minimize the data consistency error between the acquired data and the forward model described by the encoding matrix. The encoding includes the effect of the motion trajectory as well as the Fourier encoding and undersampling. The encoding matrix, \( E_{\theta} \), for the motion pa-
rameters $\theta$, is described mathematically as:

$$E_\theta = UFCTT_\theta R_\theta$$

(4.2)

where $R_\theta$ is the rotation operator, $T_\theta$ is the translation operator, $C$ applies coil sensitivities, $F$ is Fourier encoding, and $U$ is the sampling operator. The motion parameters are found by minimizing the data consistency error between the acquired k-space data, $s$, and the k-space data generated by applying the motion forward model $E_\theta$ to the CNN image:

$$[\hat{\theta}] = \text{argmin}_{\theta} \|s - E_\theta x_{cnn}\|_2$$

(4.3)

The minimization is performed using a quasi-Newton search available with the built-in `fminunc` function in MATLAB.

As discussed in prior work\textsuperscript{62} and in Chapter 3, the underlying image and motion parameters are tightly coupled which prohibits the separation of the optimization variables into orthogonal subsets. This directly limits the performance of alternating methods to only 1 or 2 productive steps during each alternating pass. However, we will demonstrate that the application of the CNN allows for the efficient decomposition of the optimization. Specifically, the motion parameters, $\theta$, which are typically optimized in a single cost function as shown in Eqn. 4.3, can be separated to create a set of much smaller optimization problems, where we can independently estimate $\theta$ for each shot of the RARE sequence (11 k-space lines in our case). The motion parameters can then be indexed by the shot number; $\theta = [\theta_1, \theta_2, \ldots, \theta_{N_{sh}}]$ where $N_{sh}$ is the total number of shots, and each vector $\theta_n$ contains the six rigid body motion parameters for a given shot. The motion forward model, $E_\theta$, is reduced to only generate the subset of k-space associated with shot $n$, and is denoted as $E_{\theta_n}$.

Similarly, the acquired data for a single shot is denoted as $s_n$, and the cost function for a single shot
is:

\[
\hat{\theta}_n = \arg\min_{\theta_n} \| s_n - E_{\theta_n} x_{\text{cnn}} \|_2
\]

By using Eqn. 4.4 instead of Eqn. 4.3, the number of unknowns for any optimization is decreased by a factor of \(N_{sh}\) (the total number of shots). These separate minimizations can be done in parallel, which greatly improves the computational scalability of the retrospective motion correction approach.

This computational efficiency allows us to consider further refinement of the optimization variables. Namely, we extend cost function Eqn. 4.4 to consider motion within the RARE shots (intra-shot motion) by assigning additional motion parameters to the individual lines of k-space within a shot. Thus \(\theta_n\) is expanded from size 6x1 to size \((6 \times \text{ETL}) \times 1\), and can be written as \(\theta_n = [\theta_{n,1}, \theta_{n,2}, \ldots, \theta_{n,L}]\) where \(L\) is the number of lines per shot (the ETL of the RARE sequence) and \(\theta_{n,l}\) contains the six rigid body motion parameters for shot \(n\), line \(l\). The forward model is further reduced to \(E_{\theta_{n,l}}\) to generate the k-space data only for line \(l\) of shot \(n\), and the signal for that line is written as \(s_{n,l}\). To find the optimal motion parameters for an individual line, the cost function becomes:

\[
\hat{\theta}_{n,l} = \arg\min_{\theta_{n,l}} \| s_{n,l} - E_{\theta_{n,l}} x_{\text{cnn}} \|_2
\]

The ability to selectively refine the optimization variables is extremely valuable as only small portions of the acquisition are likely to suffer from significant intra-shot motion. These shots show significant data consistency error and are often considered as outliers to be discarded\(^{49}\). However, with the separable model approach we can expand the model dynamically. The parallel optimizations in Eqn. 4.4 can be performed for each shot, and then shots that had large data inconsistency after this first pass can then be improved upon using Eqn. 4.5 (see Fig. 4.3). By using this multiscale optimization for the motion parameters, the length of \(\theta\) can vary depending on how many shots require intra-shot correction, with the minimum length being 6N and a maximum length of 6NL.
Figure 4.3: How to determine if intra-shot correction is needed. A. Two different subject were imaged and asked to shake their heads in a “no” pattern to generate in-plane motion artifacts. B. NAMER was performed on both subjects, only correcting for between shot motion (i.e., assuming no motion during the echo train readout). C. The data consistency errors were evaluated after inter-shot correction, and if high data consistency shots were present, intra-shot correction was performed. If not, the image from between shot correction is returned. D. Final NAMER image for subject 1 and subject 2.

Similar to Eqn. 4.3, a quasi-Newton search using MATLAB’s fminunc is performed for equations 4.4 and 4.5.

4.1.5 Model-based image reconstruction using CNN-guided motion estimate

Using the raw k-space data, $s$, and the current estimate for the motion trajectory, $\theta^*$, the linear least squares image reconstruction problem is solved using the conjugate gradient method to find the image, $x$:

$$[\hat{x}] = \arg\min_x ||s - E_{\theta^*}x||_2$$  \hspace{1cm} (4.6)
The model-based reconstruction, \( \hat{x} \), can then be fed back through the CNN to identify remaining artifacts. The three steps (apply CNN, motion search, solve for image) are repeated until a stopping criterion is met (either a maximum number of iterations or the change in \( \hat{x} \) is below a given threshold). In all cases, the final image returned by NAMER is the model-based reconstruction from Eqn. 4.6.

4.2 NAMER experiments

4.2.1 Simulations

The performance of the motion detecting CNN was tested using simulated motion corrupted data. For an unseen subject, each slice was motion corrupted and passed through the CNN. Image space RMSE was calculated for all the motion corrupted slices. Next, a single slice was chosen from the imaging volume and NAMER was performed using both a single cost function (Eqn. 4.3), and a separable cost function (Eqn. 4.4 for all shots). Motion correction using an alternating minimization (only the 2\(^{nd}\) and 3\(^{rd}\) steps of the NAMER loop) was also performed with both a single and separable cost function, in order to compare to previous methods. No intra-shot motion corruption or correction was used for the comparison experiment. We performed 20 total iterations for all four methods, and the algorithm convergence and final images are compared. Additionally, a simulation using NAMER with the separable cost function was performed in a case with intra-shot motion corrupted data.

We also investigate the potential of combining NAMER with the reduced modeling strategies developed in Chapter 3 in a simulation experiment. During the third step of NAMER (update of the voxel values), we only updated a strip of voxels that corresponded to the width of a single patch (51 voxels), and then passed the partially updated image through the CNN at the next step. We then evaluated the total number of voxel updates required using a full volume solve versus a targeted
voxel update during step 3 of NAMER.

### 4.2.2 In vivo experiments

NAMER was applied to a single brain volume from a healthy subject who was instructed to move during the scan. The subject was asked to shake their head in a “no” pattern for a few seconds, approximately halfway through the 2-minute scan. This motion pattern was used in order to restrict motion artifacts to those caused by within slice motion. Here, a 2D implementation of NAMER, correcting for in-plane translations and rotations, was employed. First, eight iterations of NAMER were performed for each slice, assuming motion only occurred between the shots. Next, one iteration of line-by-line motion correction was performed for the three shots that occurred during the middle of the scan (shots 10, 11, and 12 out of 17 total) and had larger data consistency error.

### 4.3 NAMER results

We present results demonstrating the motion mitigation capabilities of NAMER in simulations and a supervised motion experiment. First, we show that the CNN removes motion artifacts and decreases image space RMSE in all slices of an unseen subject with simulated motion corruption. Next, we show in simulations that NAMER removes more motion artifacts than an alternating minimization for both a single cost function (Eqn. 4.3) and a separable cost function (Eqn. 4.4). Unlike the alternating method, the performance of NAMER does not suffer when using a separable cost function. Finally, NAMER removes inter-shot and intra-shot motion artifacts from an in vivo motion experiment where the subject was instructed to move during the scan.
Figure 4.4: CNN artifact mitigation across whole brain. A. Representative slices of CNN motion artifact mitigation across the brain volume for a simulated motion example. Bottom right shows the image space RMSE compared to the ground truth image. B. For all slices in the brain volume, the image space RMSE decreased after the CNN, with an average improvement of 3.9%. C. Despite large reductions in artifacts shown in A., compared to the ground truth ringing and blurring artifacts still remain.

4.3.1 Simulations

Figure 4.4 shows the artifact removal performance of the CNN when applied to simulated motion corruption of data from a subject not included in the training data. Significant ringing artifacts are removed for all slices. When compared to the ground truth images, the simulated motion resulted in an average RMSE of 20% across the slices. The average error was reduced to 16.1% through the application of the CNN. As can be seen in Fig. 4.4-C, the network removes a significant portion of the motion artifacts, but some residual ringing and blurring are present.

Figure 4.5 shows the performance of NAMER motion correction compared to an alternating method. Each approach was applied to simulated motion corruption which produced a 43.8% image space RMSE (compared to the ground truth). The final images returned by NAMER for the single cost function (Eqn. 4.3) and separable cost function (Eqn. 4.4) formulations show negli-
Figure 4.5: NAMER compared to alternating method in simulated motion data. A. Original ground truth image and simulated motion corrupted image. B. Image results after 20 iterations of NAMER or an alternating motion correction method. The white numbers in the bottom right are image space RMSE compared to the ground truth image. Top row shows the reconstruction results when using a single cost function for the motion minimization (Eqn. ??), bottom row shows results from a separable cost function for the motion minimization (Eqn. ??). C. Convergence of the four methods displayed in B are shown. Both of the NAMER implementations converge more quickly, and to a lower final data consistency error.

ble remaining artifacts and the image space RMSE decreased to 12.7% and 11.7% respectively. For the alternating method, significant artifacts remain after the 20 optimization steps. The lack of separability of the alternating optimization is observed as the single cost function outperformed the separable cost function. In that case, the single cost function produced cleaner images with a lower final RMSE (28.2% compared to 31.2%).

The convergence rates of the NAMER and alternating optimizations can be seen in Fig. 4.5-C. The initial data consistency error of the motion corrupted image was 17.4% (corresponding to 43.8% image space RMSE), compared to a data consistency error of 9.4% when using the ground truth motion. With NAMER motion correction, the data consistency error was reduced to the ground truth level of 9.4% after 9 iterations for both the single cost function (Eqn. 4.3) and separable cost
functions (Eqn. 4.4). After 20 iterations, the alternating method was only able to achieve data consistency errors of 10.4% for the single cost function, and 11.6% for the separable cost function.

For the simulated motion corruption example shown in Fig. 4.5, using a reduced model as done in Chapter 3 for image reconstruction (instead of a full volume solve at each NAMER iteration) requires 33% fewer voxels updates. Additionally, NAMER simulations of intra-shot motion correction also produced an artifact mitigated image (4.6).
Figure 4.7: NAMER in vivo inter- and intra-shot motion correction. Reconstructed images assuming no motion occurred, correcting for motion between shots (inter-shot correction), and correction results after allowing fine-tuning of the motion parameters for each line of k-space at highly corrupted shots (intra-shot correction). Motion artifacts are significantly reduced by allowing the motion parameters to vary across the lines within a shot, as shown in the right column. Data consistency error values are shown in the bottom right of each image.

4.3.2 In vivo imaging

NAMER motion correction was applied across the full 3D brain volume to correct artifacts produced by the subject shaking their head “no” during the acquisition. Slice 14 from the 35 slice stack is shown in Fig. 4.7. The image without any motion correction contains substantial ringing artifacts. Using only between shot NAMER correction with a separable cost function (Eqn. 4.4) resulted in residual ringing. By refining the model to optimize individual lines within the shots that contain the largest data consistency error (Eqn. 4.5), the ringing artifacts are significantly reduced. The data consistency RMSE (shown in bottom right of the images in Fig. 4.7) decreased for all slices within the volume. For slices within the brain region there was an average reduction of 0.2% between the no correction reconstruction and the within shot motion correction. Similar qualitative improvements in image quality were observed for the slices not shown in Fig. 4.7, and two additional slices are presented in Fig. 4.8.

For a single slice of real motion corrupted data (subject 2 in Fig. 4.3), the first step of NAMER
Figure 4.8: Full volume NAMER motion correction. A. NAMER motion correction results for three slices corrupted by subject motion during the acquisition. Left column shows the reconstruction with no motion correction, middle shows the results of NAMER motion correction when only correcting for between shot motion (i.e. assuming no motion during the readout), and the right column shows the results after allowing fine-tuning of the motion parameters for each line of k-space at highly corrupted shots. Bottom right shows data consistency error. B. Zoomed in portions of images from part A. Ringing is reduced in slices 14 and 18, and in slice 29 blurring is reduced using NAMER. All other shots in the volume achieved similar levels of motion correction.

(CNN evaluation) took on average (averaged across the iterations) 10s on a 12GB NVIDIA Tesla P100. The I/O time to interface the CNN to MATLAB required an additional 40s. The second step of NAMER (motion optimization) took on average 2.7 minutes, and the final step of NAMER (full image solve with motion parameters included) took on average 3.6 minutes.
In this work we present and validate two complementary retrospective motion correction techniques: TArgeted Motion Estimation and Reduction (TAMER) and Network Accelerated Motion Estimation and Reduction (NAMER). A brief summary of each method and their results are detailed in the subsections below.
5.1.1 Image space model reduction using the TAMER method

TAMER is a reduced model joint optimization methods that corrects for patient motion during MRI scans. TAMER retrospectively corrects for motion during reconstruction by minimizing the data consistency error between the acquired data and a SENSE plus motion forward model. TAMER relies on information about the motion trajectory that has been encoded by the coil array into the k-space data. By minimizing the data consistency, we do not rely on proxy metrics for image quality, such as gradient entropy or total variation, to drive our motion search. In order to efficiently perform this joint minimization, we have introduced a reduced model, where we perform only small reconstructions of target voxel subsets while searching for the motion parameters. TAMER only operates on the full volume data twice (image initialization and final reconstruction). This significantly reduces the computation of the joint optimization (motion parameters and image voxels).

Since TAMER is a retrospective approach which does not alter the acquisition procedure, it has the potential to be easily integrated into current clinical MRI scans. Here we presented results of the TAMER optimization applied to 2D RARE images, one of the most common clinical scans. Our approach can be extended to several other commonly used 2D and 3D clinical sequences. Unlike prospective motion correction techniques\(^\text{x}\), TAMER requires no modifications to the acquisition pulse sequence (gradient updates and navigators) nor to the clinical workflow (motion tracking hardware). Nonetheless, if data motion tracking hardware or MR navigator information is available, it could be incorporated into the TAMER optimization. Finally, this retrospective technique will allow for a direct comparison of the TAMER reconstructed image to the standard reconstruction, which is not possible when using prospective correction.

By construction, TAMER requires multicoil data (even for a fully sampled k-space acquisition) since the motion parameters cannot be found without the additional degrees of freedom afforded
through multicoil acquisition, and much of the motion information itself likely comes from the intensity and phase patterns incurred from motion within the fixed detector array. In this work we assumed static coil profiles that do not change with patient motion due to changes in coil loading, but for larger motion examples the model could be extended to include dynamic coil profiles as the subject moves through the field of view.

The objective function of our reduced model in Eqn. 3.4 can be modified by adding a weighting to the $\ell_2$-norm or by adding spatial smoothness penalties (as utilized in compressed sensing). In this work, we employed a weighting on the $\ell_2$ norm to reconstruct our rotation phantom data (Fig. 3.10), where after a first pass of an unweighted objective function, we performed a second pass where we weighted the higher frequency regions of k-space more heavily. This was done in an attempt to promote high frequency information in the optimization, as the initial result from TAMER did not adequately resolve fine structures. The weighting was not necessary for other datasets, and we attribute this to the spectral content of this object and the large amount of rotational motion applied in the experiment. Further work must be done to automate this feature of TAMER in future versions.

The main benefit of the more accurate search direction provided with a reduced model (see Fig. 3.6) will be a more efficient search of the motion parameters, but it will not guarantee a more accurate final motion estimate. In Fig. 3.12 both TAMER and the code provided Cordero-Grande, et al. 2018 \(^{41}\) ran until convergence, and the final image quality appears very similar. The fact that TAMER completely eliminates motion artifacts in synthetically corrupted data such as Fig. 3.5, where gold standard data was corrupted using the same motion forward model used in the motion reconstruction, suggests that residual motion artifacts in the \textit{in vivo} motion case, or moving phantoms, arises from an incomplete forward model. There are several sources that can contribute to residual data consistency error, which are discussed in the Future Directions section below.

Our target voxel approach using the correlation matrix showed the fastest convergence compared
to other voxel selection methods, however we would expect that due to the strong coupling along
the phase encode direction that the rotated correlation matrix of Fig. 3.7 would not perform well.
It seems that all of the patterns which provided rapid convergence also overlapped with areas of
elevated data consistency error.

5.1.2 Accelerated motion correction using the NAMER method

NAMER is a scalable retrospective motion correction method that effectively integrates a motion
artifact detecting CNN within a model-based motion estimation framework. The image estimates
provided by the CNN allow for separation of the motion parameter search into either individual
shots (containing all k-space lines in our case) or individual k-space lines. The small optimization
problems can be efficiently computed in an embarrassingly parallel fashion. This results in a highly
scalable algorithm that has the potential for clinical acceptance. In addition, the separability of the
motion optimization facilitates efficient refinement of the model to consider motion disturbances
that can occur within a shot. These specific shots can be clearly identified due to their large data con-
sistency error and line-by-line motion correction can be applied. The separability afforded by our
method allows us to focus on further improving these limited number of troubled regions without
incurring a substantial computational burden. The benefits of this model refinement can be clearly
observed (e.g., Fig. 4.7).

The model-based reconstruction used in NAMER also relaxes concerns about the accuracy, ro-
 bustness, and predictability of the CNN. As can be seen in Fig. 4.4, the CNN is able to reduce mo-
tion artifacts across all slices from a previously unobserved data set. But, following our expectations,
the CNN is not able to completely remove all of the artifacts (see Fig. 4.4, slice 20) and it can intro-
duce undesirable blurring to the images (see Fig. 4.4, slice 14). However, the CNN output is accu-
rate enough to both improve the convergence of the reconstruction and promote the separability
of the motion parameter optimization. Through the inclusion of more diverse and larger training
data sets we expect these benefits to grow. In addition, the NAMER method presented in this work utilizes a standard convolution network, though further improvements might be achieved with an optimization network architecture (see Future Directions).

The NAMER motion mitigation method was assessed through both simulations and supervised in vivo motion experiments. These include testing the CNN artifact removal capabilities on simulated 2D motion corrupted data, simulations to show convergence of the method toward a ground truth, and the supervised head shaking experiment where 2D motion occurred parallel to the imaging planes. The reconstruction framework presented here does generalize to 3D motion trajectories and we think similar benefits can be achieved with adequate training of the CNN, although these remain to be demonstrated.

5.2 Future Directions

There are many future directions for this work, with the primary areas being the full combination of the TAMER and NAMER approaches, improved handling of 3D (i.e., through-plane) motion, motion forward model improvements, and implementation improvements. Each of these areas is discussed in more detail in the sections below.

5.2.1 Combination of TAMER and NAMER approaches

The TAMER method (Chapter 3) can be more fully combined with the NAMER method (Chapter 4). The NAMER method currently passes the whole image through the CNN at the first step, and resolves for the entire image at the final step. However, this could be altered to only correction a targeted subset of voxels at each step, as done in the TAMER method. Replacing the third step of the NAMER algorithm (see Fig. 4.1) with a reduced model image reconstruction could improve algorithm efficiency and in this case the CNN would only need to evaluate the patches that con-
tain updated voxel values. Preliminary simulation results presented in Section 4.3.1 demonstrate the potential of the approach, where 33% fewer voxel updates were required for the same motion correction performance.

5.2.2 Improvements for 3D Motion Correction

Figure 3.12 shows an example of 3D motion correction using the TAMER method, but we have yet to demonstrate 3D results using the NAMER method and in a combined TAMER/NAMER method there are improvements that should be considered for 3D motion correction.

For 3D motion correction with NAMER, the 2D CNN developed here can still be used to mitigate some motion artifacts. However, we think training data that incorporates through-plane motion effects will increase the performance of the CNN in general patient motion situations. Due to the limited range of rigid body brain motion, we anticipate that a CNN could be trained on small stacks of slices, where the objective is to remove artifacts from the interior voxels in the stack. The challenges we anticipate in extending NAMER to 3D are present in both the CNN step and the model-based motion optimization. First, the performance of the CNN on data corrupted by through-plane motion that includes spin history effects (which are currently not in the simulated training data model) would need to be evaluated. Second, previous model-based motion optimizations did not include spin history effects into their motion models, and this could hinder NAMER’s performance even if the CNN was able to filter out many of these effects.

Prior works have also shown that outlier rejection or soft gating can be beneficial for the mitigation of through plane motion artifacts, and we see those prior works as complementary to the NAMER method. The outlier rejection strategy from those works could be used to determine which shots need to have intra-shot motion correction (shown in Fig 4.7). Further, the data interpolation step in prior work could instead be replaced with our motion mitigating CNN, similar to the cascading network approach. This could potentially relax the requirement of 2x oversampling.
used in Cordero-Grande, et al. 2018\(^1\) to achieve robust through plane motion correction.

For ease of implementation we considered the target voxel regions in TAMER to be uniform across all slices in this work. Thus, the circular target voxel subsets shown for a single slice example in Fig. 3.7 were transformed to cylinders extending across all slices for a multi-slice imaging volume (such as in Fig. 3.12). However, it is likely to be more efficient to use smaller target voxel subsets that consider the restricted nature of voxel coupling in the through plane or 3D direction, in which case the target voxel subsets would be shortened in the slice direction, for example, shorter cylinders (only across a subset of slices) or spheres.

5.2.3 Motion forward model improvements

In the case of standard parallel imaging, there can be error due to sensitivity map inaccuracies and noise in the individual channel data. Depending on the image SNR this alone can be over 5%. In the presence of motion, additional model inaccuracies can contribute to final data consistency RMSE level. The most likely sources of model error for the remaining artifacts are through-plane motion in the cases where it was not included in the model, spin history effects, and intra-shot motion. In future work, more sophisticated motion models are likely needed to allow full motion mitigation without requiring ancillary techniques which go beyond data consistency, such as the acquisition of oversampled data, reliance on outlier rejection, or regularization or assumptions of through-plane smoothness.

5.2.4 Implementation improvements

An important area of future work will be to create an efficient implementation of the TAMER algorithm. Our current implementation relies on standard MATLAB motion operations and optimization algorithms, which significantly increases overall computation time. We think it could be
beneficial to incorporate implementation strategies used in Cordero-Grande, et al. 2018⁶¹, such as
analytical gradient calculations for the motion, a custom non-linear optimization algorithm, and
employment of GPU based computation.

5.2.5 Convolutional neural network optimization

Improvement of the CNN used in NAMER might be achieved with more sophisticated networks
or loss functions⁶⁹. The topology used here could benefit from further optimization of design pa-
rameters, where a sensitivity analysis across network attributes could be performed (i.e., patch size,
number of hidden layers). It is important to note that the optimal network may not necessarily be
the one which has the lowest validation loss during network training, but instead will be the net-
work that best aids in advancing the motion optimization (step 2 of NAMER).

5.3 Conclusion

This thesis presents two methods for retrospective motion correction of MRI images, TA-
rgeted Motion Estimation and Reduction (TAMER) and Network Accelerated Motion Estimation and
Reduction (NAMER). These two methods allow for motion correction without the use of exter-
nal hardware or sequence modifications, and have been validated in simulations, phantom imaging,
and human subject imaging. They also demonstrate how various mathematical and computational
techniques, such as joint optimization, reduced modelling, and convolutional neural networks, can
be applied to the problem of reconstructing MRI images in the presence of patient motion. With
future improvements in the handling of 3D (through plane) motion and in creating an efficient on
scanner implementation, the TAMER and NAMER methods will facilitate the application of re-
troractive motion correction to clinical brain images. This has the potential reduce the number of
repeated MRI scans, reduce the number of patient callbacks, and to produce overall higher quality
images that result in better diagnosis and treatment of neurological diseases.
References


This thesis was typeset using \LaTeX, originally developed by Leslie Lamport and based on Donald Knuth’s \TeX. The body text is set in 11 point Egenolf-Berner Garamond, a revival of Claude Garamont’s humanist typeface. A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu.