Efficient Algorithms for Retrospective Motion Correction in MRI

by

Yuan Bian

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 3, 2019

Keywords: MRI motion correction, retrospective motion correction, GANs, image registration, image reconstruction

Copyright 2019 by Yuan Bian

Approved by

Stanley J. Reeves, Chair, Professor of Electrical and Computer Engineering
Thomas S. Denney Jr., Professor of Electrical and Computer Engineering
Gopi Deshpande, Associate Professor of Electrical and Computer Engineering
Dmitry Glotov, Associate Professor of Mathematics and Statistics
George T. Flowers, Professor and Dean of Graduate School
Abstract

Compared with other medical imaging modalities, MR imaging is time consuming due to the relatively slow sequential data acquisition pattern. Hence, motion is often an unavoidable issue for MRI. Object motion during the signal acquisition can reduce image quality due to the induced artifacts, which further hinders diagnosis and scientific research. These degraded images may require repeated scans, which leads to treatment delay and cost increases. If images with motion artifacts are not properly interpreted, erroneous diagnoses and false scientific findings may occur.

To reduce motion artifacts, three groups of methods are used by practitioners: motion prevention, artifact mitigation, and motion correction. Motion prevention methods are the most straightforward way to reduce motion artifacts. However, these may not always be suitable or effective. Artifact mitigation methods mainly include faster imaging and periodic triggering and gating. Imaging speed has limits, and triggering and gating require extra time, effort, and complexity. Therefore, motion correction methods have received significant attention.

MRI motion correction techniques can be classified into three groups: motion correction based on k-space trajectories, prospective motion correction (PMC) and retrospective motion correction (RMC). Motion correction based on k-space trajectories relies on specially designed and implemented trajectories, which limits the flexibility of the techniques and requires more acquisition time. PMC is achieved by obtaining tracking data of the pose (position and orientation) of the object, passing these data to the scanner with minimal delay, and adjusting the MR pulse sequences so that the imaging volume moves to follow the object. PMC requires extra hardware and calibration and sometimes extra acquisition
time. RMC postprocesses the data and reconstructs MR images after the data is fully acquired. In RMC the process of acquisition is independent of motion. RMC includes three main groups: self-navigation motion tracking methods to calculate motion information, autofocusing methods based on evaluation of image quality, and motion correction by training neural networks. Self-navigation motion correction methods rely on Fourier properties by taking advantage of overlapping k-space data to track motion. This approach requires additional k-trajectories, which increases both time and complexity of the scan. Autofocusing methods do not rely on a specific data sample pattern, equipment or sequence design. These approaches assume a rigid body or deformable object motion model and estimate motion model parameters by iterative optimization of an image quality metric when the raw k-space data are modified according to the motion model. Artificial neural network methods establish the mapping relationship between the motion-corrupted images and the no-motion images by training a large number of related images, and estimate motion-corrected images from motion-corrupted images. The focus of this thesis is the development of autofocusing and neural-network approaches to RMC.

In this dissertation, we develop three methods to correct MRI motion retrospectively. The first contribution is an autofocusing motion correction method to address the two challenges of previous methods: high calculation load and local minima. We propose to use multiple linear-motion initializations and joint refinement of a global model to decrease and constrain the search space. In the first step, k-space is divided into several segments based on acquisition order. Linear motion is assumed and searched in each segment to get initial motion parameters. In the second step, several control points are chosen on the piecewise linear initial approximation, and then a piecewise cubic Hermite interpolation polynomial is fitted from the control points to obtain smooth motion curves. The motion curves are refined by optimizing a focus criterion. These strategies make the proposed algorithm efficient and robust. Different focus criteria are compared under the proposed method. To further improve computational efficiency, golden-section search is used to estimate rotation, and
two map data structures are applied to store calculated data. Simulations and experimental results demonstrate that the proposed method can effectively and efficiently correct rigid motion in MR images.

The second contribution of this work is an efficient motion correction method based on fast robust correlation. Translational search can be computationally demanding. A correlation operation can be used to calculate an image match when the matching criterion is the sum of squared errors. However, this approach cannot be used for nonquadratic matching criteria. Fast robust correlation is a computationally efficient search algorithm for translational image matching in the frequency domain. This method can calculate matching surfaces from nonquadratic criteria using a series of high-speed correlations by defining a kernel with sinusoidal terms. The proposed method corrects motion-distorted images by aligning translational motion between images formed by neighboring frequency segments. Since the squared difference kernel is invariant to motion between partial-Fourier images, we adopt the absolute value kernel, which can be easily approximated by sinusoidal terms. Total variation of the sum of partial-Fourier images is chosen as the new matching criterion. FFTs are used to calculate correlations for computational speed. Different search strategies to combine and correct motion over the whole k-space are discussed and compared. The proposed method can perform real-time processing to reduce image motion artifacts significantly in the simulations and MRI cardiac experiments.

The third contribution of this dissertation is a novel data-driven motion correction method for magnitude MR images using generative adversarial networks (GANs). Although the previous proposed methods can correct motion effectively and efficiently, they both require complex-valued raw data. However, raw data is not usually preserved in a clinical environment. In this case the previous two methods cannot be used. GANs (pix2pix model) are implemented to reduce motion artifacts and reconstruct motion-corrupted images through adversarial training between generator and discriminator networks to estimate a motion-corrected image that is close to the reference image. The training set is made of
image pairs consisting of motionless reference images and corresponding motion-simulated images. The proposed method is validated by a simulated motion test set and a real motion (experimental) test set.
Acknowledgments

I would like to thank my advisor, Dr. Stanley J. Reeves, for his encouragement and guidance over the years. Dr. Reeves supported me throughout my research with his patience while sharing me with his insightful ideas. This dissertation would never have existed without him.

I would like to thank Dr. Thomas S. Denney, Dr. Gopikrishna Deshpande, and Dr. Meredith Reid, for providing great suggestions in my research. I would like to thank Dr. Dmitry Glotov, who provided invaluable comments for this work. I would like to thank Dr. Ronald Beyers for his help of experimental preparation and MRI data acquisition. I also thank Dr. Yanzhao Cao for serving as the outside reader for this dissertation.

I would like to thank all my group members, especially Xiaoxia Zhang, Sinan Zhao, Yu Shao, Chenxi Hu, Ming Li, Hai Lu, Shuo Shang, and Huanyi Zhou, for their kindness and suggestions during my research.

Finally, I would like to thank my beloved parents, Jinyue Bian and Guang Shao, for the endless love they gave me. They always encourage me and support my decisions in each step of my life. The have always taken care of me no matter where I was.
# Table of Contents

Abstract .................................................. ii  
Acknowledgments ........................................ vi  
List of Figures ............................................... x  
List of Tables ............................................... xx  

1 Introduction ........................................... 1  
1.1 MRI basics ........................................... 1  
1.1.1 MRI physics ....................................... 1  
1.1.2 MR image formation ................................. 2  
1.2 MRI motion ........................................... 4  
1.3 Overview of the dissertation ......................... 6  

2 MRI Motion Correction ................................. 8  
2.1 Motion correction based on k-space trajectories ...... 8  
2.2 Prospective motion correction ......................... 9  
2.2.1 motion correction strategy ....................... 10  
2.2.2 motion tracking ................................... 11  
2.3 PMC methods ......................................... 13  
2.3.1 PACE and PROMO ................................. 13  
2.3.2 SNAV ............................................ 14  
2.4 Retrospective motion correction .................... 15  
2.5 Autofocusing methods ................................. 16  
2.5.1 automatic correction of motion artifacts in Magnetic Resonance images  
using an entropy focus criterion ............................ 16  
2.5.2 blind retrospective motion correction of MR images .............. 18
List of Figures

1.1 Excitation process. Left: The net magnetization is parallel to the external strong magnetic field $B_0$ before an RF pulse. Center and right: An RF pulse with the resonant frequency allows the RF energy to be absorbed by the protons, and makes the net magnetization to rotate away from $B_0$. (This image is from [80])

1.2 Three gradient coils are inside the bore of an MR scanner. (This image is from the link https://slideplayer.com/slide/9887251/)

2.1 Trajectories of a PROPELLER composed of 9 strips. (Figure 2.1 is from [77].)

2.2 The strategy of PMC.

2.3 Gradients change with rotation of the head. (Figure 2.3 is from [64].)

2.4 Process of PACE.

2.5 3D IR-SPGR sequence with SP-Navs and EKF. (Figure 2.5 is from [100].)

2.6 (a) Signal amplitude of spherical shell (b) before and (c) after a 12° relative rotation. (Figure 2.6 is from [98].)

2.7 Optimization strategy of Blind Retrospective Motion Correction of MR Images. The k-space data is processed from center (low frequency part) to the whole k-space to optimize motion parameters using coarse to fine optimization. (Figure 2.7 is from [59].)
2.8 (a) Sample locations after RMC correction for rotation object. (b) Sample locations after PMC for rotation object. ................................................. 23

3.1 Different times brain images registration to detect a lesion. (This image is from the link https://thebestphotos.eu/detection-of-focal-longitudinal-changes-in-the-brain.html) ................................................................. 26

3.2 Different views aerial images registration to extend the view. (This image is from MathWorks website Image Registration section, and the link is https://www.mathworks.com/discovery/image-registration.html) ................. 26

3.3 Different sensors brain images registration to fuse images. (This image is from UCF Medical Image Computing (CAP 5937) slices (Spring 2016), and the link is http://www.cs.ucf.edu/\protect\unhbox\voidb@x\penalty\@M\{}bagci/teaching/mic16/lec15.pdf) ................................................................. 26

3.4 The steps of area-based registration methods ........................................ 27

3.5 The steps of feature-based registration methods .................................... 27

3.6 Affine transformation model, the parallel lines keep parallel after an affine transformation. ................................................................. 30

3.7 A thin-plate spline that passes through a set of control points. (Figure 3.7 is from the link http://elonen.iki.fi/code/tpsdemo/.) .......................... 31

3.8 Left: source image. Middle: target image. Right: deformed source image through B-splines registration. (Figure 3.8 is from [103].) .......................... 31

4.1 The motion model of the proposed method. in k-space, each dash line is one $k_{\text{phase}}$ line. Different $k_{\text{phase}}$ lines have their own motion parameters $(\Delta x, \Delta y, \Delta r)$. Red, green and blue are three $k_{\text{phase}}$ lines with three different sets of motion parameters. 39
4.2 The strong relationship between neighboring lines results in motion estimation error. Segment 1 and segment 2 are two neighboring segments in k-space and the green solid line gives the real motion for each k-space line of the segment 1 and the segment 2. Due to the strong bond between the red and blue thick solid lines, the estimated motion of the segment 1 is dominated by the red thick solid line, which results in the errors for other lines in the segment 1.

4.3 The process of piecewise linear initial approximation. There are three motion curves with the $x$ axis representing $k_{\text{phase}}$ lines and the $y$ axis representing pixels for $x$, $y$ translations and rotation degree for rotation. The whole k-space is divided into three segments with a certain number of $k_{\text{phase}}$ lines in each segment. In Segment 1, the linear estimations of the three motions are chosen from 729 (9 $\times$ 9, in each motion curve, 9 slopes (blue dashed line) are given) different motion combinations, which can minimize the criterion of the image formed by Segment 1, marked as red lines. In Segment 2, the linear estimations of the three motions are chosen from 729 different motion combinations all starting from the linear estimation of Segment 1, which can minimize the criterion of the image formed by Segment 1 and Segment 2, marked as green lines. The process is repeated for Segment 3.
4.4 The process of joint refinement of global model. In the first row, four equally spaced control points (black cross) are chosen on the piecewise linear approximation (red linear lines) for each motion parameter, and the globally criterion optimization is accomplished by refining three piecewise cubic Hermite interpolating polynomials (three green dashed curves) fitted from the four control points. One more control point is added from the previous refined motion curves (red curves in the second row) to get a refined curves (green dashed curves in the second row). The process is repeated until the desired model precision is accomplished (three red curves in the third row). ................................................................. 46

4.5 The process of rotation correction. The frequency domain images are logarithmically scaled. ................................................................. 47

4.6 One step of golden-section search to find the minimum. (Figure 4.6 is from [101].) 48

4.7 The map is used to store the rotated k-space data with rotation angle as the key. When the rotation angle $-3.38^\circ$ is searched, the corresponding rotated k-space matrix $F_{6M\times N}$ is returned. ................................................................. 50

4.8 Rotation interpolation. The frequency domain images are logarithmically scaled. The length and width of the zeropadded images are twice the original image. . . 51
4.9 The difference images comparison. The same image is used as the reference images but different floating image. For (a), the floating image is generated by Figure 4.8. For (b), the floating image is generated by rotating the reference image directly. In order to acquire the same orientation with the reference image, both floating images rotate an angle clockwise and then rotate the same angle counterclockwise. (In order to rotate clockwise and counterclockwise, for (a), the padding zero image is used to rotate and then downsampled.) Both difference images are shown at the same scaling. The dominant error occurs around the edge in (a), while the image content error is the main error in (b) with a bigger margin.

4.10 Three different motions. The number of function values (160) corresponds to the number of $k_{\text{phase}}$ lines. Horizontal axis coordinates from 1 to 160 correspond to centric-out phase-encode lines acquisition order from the central (low frequency $k_{\text{phase}} = 0$) to the peripheral region of k-space (high frequency); vertical axis coordinates correspond to the motions of each $k_{\text{phase}}$ line in one of the three motion directions (two translations with unit pixel and rotation with unit degree). (a) piecewise linear. (b) parabolic. (c) sinusoidal.

4.11 Comparison of the reference image, motion-corrupted image, and five corrected images. (a) The reference image. (b) The motion-corrupted image. (c) The motion-corrected image based on image entropy. (d) The motion-corrected image based on the entropy of one-directional gradient of image along the phase-encoding direction. (e) The motion-corrected image based on entropy of two directional gradients of image. (f) The motion-corrected image based on total variation. (g) The motion-corrected image based on the sum of absolute value.
4.12 Comparison of the reference image, motion-corrupted image, and five motion-corrected images and corresponding motion estimation plots. (a) The reference image. (b) The simulated motion-corrupted image. (c) The motion-corrected image based on the image entropy. (e) The motion-corrected image based on the entropy of the one-directional gradient of image along the phase-encoding direction. (g) The motion-corrected image based on the entropy of the two directional gradients of image. (i) The motion-corrected image based on the total variation. (k) The motion-corrected image based on the sum of the absolute values. (d)(f)(h)(j)(l) are the comparisons of the estimated motion trajectories and real motion trajectories based on the five criteria. The solid curves represent real motion trajectories (green for horizontal motion, red for vertical motion and black for rotation), and the dashed curves represent estimated motion trajectories (green for horizontal translation and red for vertical translation with unit pixel of the y axis; black for rotation with unit degree of the y axis).
4.13 Comparison of the reference image, motion-corrupted image, and five motion-corrected images and corresponding motion estimation plots. (a) The reference image. (b) The simulated motion-corrupted image. (c) The motion-corrected image based on the image entropy. (e) The motion-corrected image based on the entropy of the one-directional gradient of image along the phase-encoding direction. (g) The motion-corrected image based on the entropy of the two directional gradients of image. (i) The motion-corrected image based on the total variation. (k) The motion-corrected image based on the sum of the absolute values. (d) (f) (h) (j) (l) are the comparisons of the estimated motion trajectories and real motion trajectories based on the five criteria. The solid curves represent real motion trajectories (green for horizontal motion, red for vertical motion and black for rotation), and the dashed curves represent estimated motion trajectories (green for horizontal translation and red for vertical translation with unit pixel of the y axis; black for rotation with unit degree of the y axis).

4.14 Comparison of the reference image, real motion-corrupted image and motion-corrected image in the sagittal plane. Due to the motion difference between the neck part and the head part, the image above the green dashed line part was used to calculate the focus criterion. (a) No-motion (reference) image. (b) Real motion-corrupted image. (c) Motion-corrected image.

4.15 Comparison of the no-motion image, real motion-corrupted image and motion-corrected image in the axial plane. (a) No-motion (reference) image. (b) Real motion-corrupted image. (c) Motion-corrected image.

5.1 The motion assumption for the proposed motion correction method. We only consider the motion between different segments. Within each segment, the object is assumed motionless.
5.2 The comparison between the filtered and unfiltered square wave. (a) The magnitude and phase of the filter. (b) The filtered and unfiltered square wave.

5.3 Comparison of different values of $P$ and the absolute value kernel. (a) $P = 1$. (b) $P = 2$. (c) $P = 3$. (d) $P = 4$. (e) $P = 5$. (d) $P = 6$.

5.4 Search Strategy 1 of the proposed method. Different numbers represent different segments. The yellow segment is the reference segment, the light blue one is the corresponding floating segment, and the blue one is the motion-corrected segment. From top to bottom, the relative motion between the reference and floating segments is determined.

5.5 Search Strategy 2 of the proposed method. Different numbers represent different segments. The blue segment is the reference segment, and the light blue one is the corresponding floating segment. The reference segment is enlarged after combining the previous corrected floating segment until all the segments are corrected.

5.6 Search Strategy 3 of the proposed method. The blue segment is the reference segment, and the light blue one is the corresponding floating segment. The reference segment is enlarged after combining the previous corrected floating segment until all the segments are corrected. The neighboring symmetric segments are in turn chosen as the floating segments.

5.7 Search Strategy 4 of the proposed method. The blue segment is the reference segment, the light blue one is the corresponding floating segment and the yellow one is the segment corrected twice. Take segment 3 for example. Segment 3 and 4 are symmetric segments. After segment 3 and segment 4 are corrected in turn, the segment 3 is corrected again with the corrected segment 4 as part of the reference.
5.8 Search Strategy 5 of the proposed method. The blue segment is the reference segment, the light blue one is the corresponding floating segment and the yellow one is the segment corrected twice. The symmetric segments are corrected twice.

5.9 Example 1 of the proposed method with segment size 8.

5.10 Example 2 of the proposed method with segment size 10.

5.11 Example 3 of the proposed method with segment size 12.

5.12 Example 4 of the proposed method with segment size 15.

5.13 Comparison of breath-hold no-motion images, original free-breathing images and reconstructed images from the original free-breathing images for two cardiac phases. The first row is one cardiac phase, and the second row is the other cardiac phase. (a)(d) breath-hold no-motion images. (b)(e) original free-breathing images. (c)(f) reconstructed images from the original free-breathing images.

6.1 One multilayer perceptron neural network. The blue neurons represent the neurons from input layer, the green neurons represent the neurons from hidden layers, and the purple neuron represent the neurons from output layer.

6.2 Example of a convolutional neural network. (Figure 6.2 is from [62].)

6.3 An example of a GAN. (Figure 6.2 is from https://skymind.ai/wiki/generative-adversarial-network-gan.)

6.4 An example of a cGAN. The discriminator $D$ is trained to distinguish the tuple between motion-corrected image $G(y)$ and motion-corrupted image $y$ from the tuple between motionless image $x$ and motion-corrupted image $y$. The generator $G$ is trained to fool the discriminator $D$. Both the generator $G$ and discriminator $D$ can access the motion corrupted image $y$ in the cGAN.
6.5 Results of motion correction on five simulation datasets from the simulation test set. For each dataset, we use two rows to compare the results. The first row consists of three images from left to right: motionless image (reference), motion-corrupted image, the absolute value of the difference between the reference and motion-corrupted image. The second row consists of three images from left to right: motionless image (reference), motion-corrected image, the absolute value of the difference between the reference and motion-corrected image.

6.6 Results of motion correction on two experimental datasets from the experimental test set. The first column shows no-motion images generated by subject motion in the experiment; the second column shows motion-corrupted images generated by motionless subject in the experiment; the third column shows the motion-corrected images through pix2pix model.
List of Tables

3.1 Search directions of different algorithms .................................................. 35

5.1 Average translational motion estimation errors (two directions $\Delta_x, \Delta_y$) of each k-space segment (8 lines) for each search strategy (pixel/unit) ......................... 80

5.2 Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (8 lines segment) 80

5.3 Average translational motion estimation errors (two directions $\Delta_x, \Delta_y$) of each k-space segment (10 lines) for each search strategy (pixel/unit) ................. 81

5.4 Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (10 lines segment) 81

5.5 Average translational motion estimation errors (two directions $\Delta_x, \Delta_y$) of each k-space segment (12 lines) for each search strategy (pixel/unit) ................. 81

5.6 Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (12 lines segment) 82

5.7 Average translational motion estimation errors (two directions $\Delta_x, \Delta_y$) of each k-space segment (15 lines) for each search strategy (pixel/unit) ................. 82

5.8 Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (15 lines segment) 82

6.1 Average values of three metrics of the motion-corrupted images and the motion-corrected images ................................................................. 100
Chapter 1
Introduction

This chapter provides the background of MRI motion, and is organized as follows: 1.1 MRI basics, 1.2 MRI motion, and 1.3 overview of the dissertation.

1.1 MRI basics

Magnetic Resonance Imaging (MRI) is a widely used medical imaging technique which exploits the phenomenon that in a magnetic field, some nuclei can absorb and then re-emit energy, which can be detected under certain conditions. MRI is non-invasive, uses non-ionizing radiation, and provides contrast in soft-tissue structures. This capability is an indispensable imaging tool both in research and in clinical applications. An MRI scanner is a complicated system. Here we only discuss the basic MRI physics and MR image formation.

1.1.1 MRI physics

The physical phenomenon of MRI is called Nuclear Magnetic Resonance. Certain nuclei have a property called nuclear spin, which is randomly oriented without an external magnetic field. When these nuclei are placed in an external strong magnetic field $B_0$, these spins are quantized to produce a net magnetization vector $M$, which is in the same direction with $B_0$. Nuclei can absorb radio-frequency (RF) energy, which can push $M$ out of alignment with $B_0$ by sending an RF pulse with a certain frequency. $M$ becomes orthogonal to $B_0$, and the nuclei precess around $B_0$. This process is called nuclear magnetic resonance or excitation, which is shown in Figure 1.1. The RF pulse is designed to match the resonant frequency of the system to resonate these nuclei. This behavior is characterized by the Larmor equation,
which is given by

\[ \omega_0 = \gamma B_0, \] (1.1)

where \( \omega_0 \) is the precession frequency of \( M \) under \( B_0 \), and \( \gamma \) is the gyromagnetic ratio. Equation (1.1) is fundamental to MRI.

Figure 1.1: Excitation process. Left: The net magnetization is parallel to the external strong magnetic field \( B_0 \) before an RF pulse. Center and right: An RF pulse with the resonant frequency allows the RF energy to be absorbed by the protons, and makes the net magnetization to rotate away from \( B_0 \). (This image is from [80])

After the RF pulse is withdrawn, \( M \) begins to return to the original equilibrium state, which is the same direction as \( B_0 \). During the return process, \( M \) continues to precess at the frequency \( \omega_0 \), and energy is emitted. A receiving coil is used to produce a signal by detecting \( M \). The generated signal has the same frequency \( \omega_0 \).

1.1.2 MR image formation

Spatial encoding is used to match the generated signal to the position of its source. Three gradient coils shown in Figure 1.2, which generate three field gradients \( G_x \), \( G_y \) and \( G_z \) in the \( x \), \( y \) and \( z \) directions, are used to achieve spatial encoding. The total magnetic field in the \( z \) direction is given by:

\[ B_z(z) = B_0 + G_z z, \] (1.2)
where $B_0$ is in the $z$ direction.

Slice selection is the first step for spatial encoding, which is used to choose a 2D plane from the 3D object to be excited. According to the Larmor equation, Eq (1.2) can be used to implement slice selection.

After slice selection, the spatial encoding becomes a 2D problem within a plane. According to Fourier Series Theory, any image can be represented as the sum of a series of 2D sinusoids. In the spatial encoding of MRI, every sample of the measured signal is the sum of all the spins in the excited slice at one time point. At each time point, the spins are excited in a wave pattern, which corresponds to a particular 2D sinusoid type, across the selected slice. In general, the $x$ direction is called the frequency encoding direction and the $y$ direction is called the phase encoding direction. The sampled domain is called k-space, which is usually sampled in the pattern of an array with the same size as the reconstructed image. The k-space sample values are complex.

According to [71], the relation between samples in k-space $S(k_x, k_y)$ and spin density $\rho(x, y)$ is given by

$$S(k_x, k_y) = \int \int \rho(x, y) \exp\left\{-i2\pi (k_x(x) + k_y(y))\right\} dx dy, \quad (1.3)$$
where \( k_x(t) \) and \( k_y(t) \) are given by

\[
k_x(t) = \frac{\gamma}{2\pi} \int_0^t G_x(\nu) d\nu, \tag{1.4}
\]

\[
k_y(t) = \frac{\gamma}{2\pi} \int_0^t G_y(\nu) d\nu. \tag{1.5}
\]

Equation (1.3) is the Fourier transform of \( \rho(x,y) \). The inverse Fourier transform of samples in k-space is the reconstructed image, which is given by

\[
f(x,y) = \int \int S(k_x, k_y) \exp \left\{ i2\pi (k_x(t)x + k_y(t)y) \right\} dk_x dk_y. \tag{1.6}
\]

### 1.2 MRI motion

Compared with other medical imaging modalities, MR imaging is time consuming due to the relatively slow sequential data acquisition pattern. Hence, motion is often an unavoidable issue for MRI. The common reasons for human motion include:

**Muscle property.** Subjects cannot keep still during the long scan time, because muscles cannot stand long time extension, especially for some gestures.

**Periodic involuntary motion.** These involuntary motions include breathing, cardiac beat, vessel pulsation and blood flow.

**Disease.** Some muscles cannot be controlled due to diseases, such as Parkinson’s disease and stroke patients [86].

**Uncontrollable subjects.** Some subjects like children [16, 79] and those with mental disorders may not stay in the scanner compliantly.

Object motion during the signal acquisition can reduce image quality due to the induced artifacts, which further hinders diagnosis and scientific research. These degraded images may require repeated scans, which leads to treatment delay and cost increases. If the artifact images are not reasonably understood, the erroneous diagnoses and false scientific findings may occur [4, 41].
In general, motion artifacts are the results of a combination of effects [108]: image sharp edge blur, ghosting, signal loss, and undesired signal appearance.

MRI motion can have different classification criteria [34]. Based on deformation, motion can be grouped into rigid or nonrigid motion. Based on time difference, motion can be separated into inter-scan and intra-scan motion. The inter-scan motion is motion that occurs between excitation pulses, while intra-scan motion is motion that occurs within an excitation pulse. Based on motion range, for 2D motion, motion within a 2D plane is called in-plane motion, while the motion out of a 2D plane is termed through-plane motion. Motion can also be divided into periodic and random motion.

In order to reduce motion artifacts, three groups of methods are implemented [108]: motion prevention, artifact mitigation, and motion correction.

Motion prevention methods are the most straightforward ways to reduce motion artifacts. Training subjects in MRI environment, scanning during the sleep of infants and babies, using anesthesia for young children [67, 47], constraining bodies by foam restraints or head holders [26], and breath-holding during cardiac or abdomen imaging are the most common strategies to prevent motion. However, these methods are not always effective [102, 67], and some of them are very uncomfortable.

Artifact mitigation methods mainly include faster imaging and periodic triggering and gating.

Faster imaging is used to reduce motion artifacts by improving imaging efficiency. Subjects are likely to keep still and feel less uncomfortable in shorter scan time. Some sequences, such as FLASH [40], bSSFP [85] and EPI, can implement faster imaging. But these methods have tradeoffs between speed and image resolution and SNR. In addition, the acquisition speed cannot be accelerated without bound, considering the practical equipment limitations and human sensation [45]. Even if the acquisition time is made shorter, some kinds of motion still cannot be avoided. Parallel imaging [88, 81, 36] and compressed sensing [63] are other tools to accelerate imaging. Coil sensitivity calibration is a key to parallel imaging.
New artifacts may be introduced if the reference data is not processed properly [38, 76]. Compressed sensing may introduce patchiness artifacts in some cases [110].

Triggering and gating are widely used in cardiac and abdominal imaging, because breath and heartbeat are quasi-periodic. Triggering method collects data at the same cardiac or breath phase of different cycles. Gating acquires data continuously and then reorders the data. Triggering and gating need extra signals and sensors, such as ECG. In addition, both methods rely on a periodicity assumption, which is not easily satisfied for arrhythmia patients and pediatrics, or other kinds of motion.

Neither motion prevention nor artifact mitigation strategies are always practical. Therefore, motion correction methods are often indispensable. We will discuss motion correction methods in Chapter 3 in detail.

1.3 Overview of the dissertation

This dissertation is structured as follows. In Chapter 2, we discuss MRI motion correction techniques, and classify them into three groups: motion correction based on k-space trajectories, prospective motion correction and retrospective motion correction. In Chapter 3, we summarize the image registration problem and discuss three aspects of it—transformation models, matching criteria, and optimization methods. In Chapter 4, we propose an efficient and robust autofocusing MR image motion correction method using global motion estimation to address the two challenges of previous methods: high calculation load and local minima. We propose to use multiple linear-motion initializations and joint refinement of a global model to decrease and constrain the search space. Different focus criteria are compared under the proposed method. To further improve computational efficiency, golden-section search is used to estimate rotation, and two map data structures are applied to store calculated data. In Chapter 5, we propose an efficient motion correction method based on fast robust correlation. The proposed method corrects motion-distorted images by aligning translational motion between images formed by neighboring frequency segments. We adopt
the absolute value kernel, which can be easily approximated by sinusoidal terms. Total variation of the sum of partial-Fourier images is chosen as the new matching criterion. FFTs are used to calculate correlations for computational speed. Different search strategies to combine and correct motion over the whole k-space are discussed and compared. In Chapter 6, we present a novel data-driven motion correction method for magnitude MR images using generative adversarial networks (GANs). GANs (pix2pix model) are implemented to reduce motion artifacts and reconstruct motion-corrupted images through adversarial training between generator and discriminator networks to estimate a motion-corrected image that is close to the reference image. The training set is made of image pairs consisting of motionless reference images and corresponding motion-simulated images. In Chapter 7, we make some observations about the research and suggest future work.
Chapter 2
MRI Motion Correction

MRI motion correction techniques can be classified into three groups: motion correction based on k-space trajectories, prospective motion correction (PMC) and retrospective motion correction (RMC). k-space trajectories are the sample patterns to fill k-space. Some k-space trajectories can enable estimating motion directly from the sample data. Prospective motion correction techniques perform a real-time update of the data acquisition strategy, which is performed by tracking object positions adaptively. Retrospective motion correction postprocesses the data and reconstructs MR images after the data is fully acquired.

This chapter is organized as follows: 2.1 motion correction based on k-space trajectories, 2.2-2.3 prospective motion correction techniques and several methods, 2.4 retrospective motion correction techniques, 2.5 autofocusing methods, 2.6 artificial neural network methods, and 2.7 advantages and limitations of PMC and RMC.

2.1 Motion correction based on k-space trajectories

Motion correction based on k-space trajectories is a type of self-navigation motion correction. PROPELLER [77] is one of the most popular k-space trajectories. PROPELLER stands for “Periodically Rotated Overlapping Parallel Lines with Enhanced Reconstruction”, which collects data in centered rectangular strips rotated around the k-space origin until filling the disc of k-space. In each strip, the same number of phase encoded lines are acquired. This sample pattern is shown in Figure 2.1.
This sample pattern can reduce motion artifacts by oversampling the k-space center. A circle in the central k-space is generated by the overlap between strips. In-plane rotations and translations of each strip are corrected by correlation between the central circle and each strip based on Fourier transform properties. Through-plane motion can be detected by cross-correlation. To reduce the effect of through-plane motion to the final reconstruction, the strips with through-plane motion are given a smaller weight factor.

The PROPELLER trajectories can be used in undersampling methods [6]. PROPELLER is very useful in clinical applications and is implemented on all the major imaging platforms. Compared to the Cartesian sampling pattern, the acquisition time of PROPELLER is increased due to the strong oversampling in central k-space. Even if PROPELLER can detect through-plane motion, it is mainly used for 2D image motion correction.

2.2 Prospective motion correction

Prospective motion correction was first proposed by Haacke and Patrick [39]. In the past thirty years, PMC has become a popular method and is widely used in MRI motion correction. The motion correction model and several kinds of motion tracking methods are shown below.
2.2.1 motion correction strategy

If an object is in motion, the object and imaging volume should maintain a constant relationship. The strategy is achieved by obtaining tracking data of the pose (position and orientation) of the object, passing these data to the scanner with minimal delay, and adjusting the MR pulse sequences so that the imaging volume moves to follow the object [64]. Figure 2.2 shows the strategy of PMC.

![Diagram of motion correction strategy]

Figure 2.2: The strategy of PMC.

Rigid motion model can be described by:

\[ x(t) = R(t)x_o + T(t), \]  

(2.1)

where \( x_o \) is the original two-element position vector before motion, \( x(t) \) is the position vector after motion occurs at the time point \( t \), \( R(t) \) is the rotation matrix of the imaged object as a function of time point \( t \) and \( T(t) \) is the translation vector as a function of time point \( t \). \( T(t) \) can be corrected by changing receive phase and RF transmit frequency. In order to compensate for \( R(t) \), the original gradient \( g_o \) should be changed to \( g(t) \) as follows:

\[ g(t) = R(t)g_o. \]  

(2.2)
The frequency, phase and slice-encoding gradients for the rotated object are combined with no motion gradients. In other words, the logical gradients, which are for the actual pose of object, are represented by combinations of the physical gradients. Figure 2.3 shows the compensation.

![Diagram of gradients change with rotation of the head.](image)

Figure 2.3: Gradients change with rotation of the head. (Figure 2.3 is from [64].)

### 2.2.2 motion tracking

In order to track the pose information of the object, three groups of tracking methods—field detection, optical methods and MR navigators—are used.

**field detection methods**

These methods were first introduced by Ackerman et al. [1]. The main idea of field detection methods is that different positions of the scanner have different magnetic field
environments, which are measured to localize the positions of the object. Active markers [74, 73] or probes [10, 11], which are attached to the subject, are required to implement the measurements. These methods have some disadvantages. The equipment is always connected by wires to the scanner, which makes the patient less comfortable and makes the scan more complicated.

*optical methods*

The use of a camera system is one of the most popular optical methods. A camera or cameras are used to track motion of markers, which are affixed to the subject, to get the position of the subject. Camera systems include out-of-bore camera system [107, 90, 5] and in-bore camera systems [83, 3, 104]. The camera is out of the scanner systems in an out-of-bore system, which reduces the compatibility requirement with the scanner but increases the challenge of accuracy. In-bore camera systems require higher MR compatibility but can generally track motion more accurately. There are two other kinds of optical methods—laser systems [28, 29] and bend-sensitive fiber—optic tape [43]. At least one marker is required in the optical system. Cross-calibration is necessary to transform the subject position information in the coordinate system of the optical tracking system to the MRI scanner coordinate system.

*MR navigators*

MR navigators include two kinds of navigators: image-based navigators and k-space navigators. Image-based navigators, such as PACE [91] and PROMO [100], make use of image registration information to update the position of the subject. These allow the user to define the interesting image region for motion correction and avoid nonrigid parts of the image region like the neck. However, the acquisition time is always longer than k-space navigators, and this long acquisition time for navigators may exceed otherwise unused time in the sequence, which may produce incompatibility problems with some MRI sequences. K-space navigators, such as orbital navigators [97], spherical navigators [98] and cloverleaf navigators [93], sample the same part of k-space data repeatedly to determine rotations and
translations by k-space comparison. Compared with field detection and optical methods, no marker and no additional equipment are needed for MR navigator methods. However, they require modified or additional k-space trajectories and may require extra time to scan the additional k-space trajectories. The information from MR navigators can also be used to correct motion retrospectively.

2.3 PMC methods

2.3.1 PACE and PROMO

Both PACE and PROMO are image-based prospective motion correction methods. Prospective Acquisition Correction (PACE) was first proposed by Thesen. This method is a combination of both prospective and retrospective motion correction. An image-based motion detection is used to estimate the 3D rigid motion, which can be used to adjust the position of the volume before the next volume acquisition. At the same time, retrospective correction is performed to decrease the residual volume-to-volume motion. Figure 2.4 shows the process of PACE.

![Figure 2.4: Process of PACE.](image-url)
PROMO represents “PROspective MOtion correction”, is another popular motion correction method based on images. Three orthogonal 2D spiral navigators (SP-Navs) are acquired to reconstruct images. The extended Kalman filter (EKF) is used to predict and correct motion based on the reconstructed images from the spiral navigators. In order to keep the scan efficient, multiple SP-Navs and EKFs are used in the recovery time of scan sequences. Figure 2.5 shows 3D IR-SPGR sequence with SP-Navs and EKFs [100].

![Figure 2.5: 3D IR-SPGR sequence with SP-Navs and EKF. (Figure 2.5 is from [100].)](image)

### 2.3.2 SNAV

A spherical shell in k-space is sampled as a 3D spherical navigator (SNAV) echo, which is used to measure rigid motion in 3D space [98]. 3D rotation and translation of an imaged object can be detected by registration in k-space. The relationship between a reference position k-space signal at the location ($k_\rho, \theta, \phi$) and a new object position at the rotated coordinate frame ($k_\rho, \theta', \phi'$) in spherical coordinates is given by:

$$S'(k_\rho, \theta', \phi') = S(k_\rho, \theta, \phi) \exp\{i2\pi k_\rho(\Delta x \cos \theta \sin \phi + \Delta y \sin \theta \sin \phi + \Delta z \cos \phi)\}, \quad (2.3)$$
where \((\Delta x, \Delta y, \Delta z)\) are the translations in the three axis directions. \(k_p\) is unchanged in the rotated coordinate, so both signals are on the same shell.

The relation between a reference position \((k_x, k_y, k_z)\) and a new translation-only object position \((k'_x, k'_y, k'_z)\) is given by:

\[
S'(k'_x, k'_y, k'_z) = S(k_x, k_y, k_z) \exp\{i2\pi(\Delta x k_x + \Delta y k_y + \Delta z k_z)\}. \tag{2.4}
\]

Since k-space registration is implemented, the three rotation parameters can be found separately from the three translation parameters. The registration can be made by rotating the magnitude values of the shell to detect the rotation information, and then translation parameters can be obtained by comparing with the phase differences after rotation is corrected. Figure 2.6 shows the spherical shell and rotation k-space registration.

![Figure 2.6](image)

Figure 2.6: (a) Signal amplitude of spherical shell (b) before and (c) after a 12° relative rotation. (Figure 2.6 is from [98].)

### 2.4 Retrospective motion correction

Retrospective motion correction refers to modifying k-space or image data during image reconstruction, after all data is acquired [34]. The process of acquisition is independent of motion. RMC includes three main groups: self-navigation motion tracking methods to calculate motion information, autofocusing methods based on evaluation of image quality, and motion correction by training neural networks. Self-navigation motion correction methods
rely on Fourier properties by taking advantage of overlapping k-space data to track motion [15, 69, 9]. Autofocusing methods do not rely on a specific data sample pattern, equipment or sequence design. These approaches assume a rigid body or deformable object motion model and estimate motion model parameters by iterative optimization of an image quality metric when the raw k-space data are modified according to the motion model [8]. Artificial neural network methods establish the mapping relationship between the motion-corrupted images and the no-motion images by training a large number of related images, and estimate motion-corrected images from motion-corrupted images.

In the next sections, we will discuss several autofocusing methods and artificial neural network methods in detail.

2.5 Autofocusing methods

2.5.1 automatic correction of motion artifacts in Magnetic Resonance images using an entropy focus criterion

To our knowledge, the first autofocusing MRI motion correction method was proposed by Atkinson et al. [8]. An image entropy focus criterion was used to correct motion-corrupted MR images. Patient motion can be described by a motion trajectory, which is assumed to be rigid motion as a function of time. Continuous movement is discretized as a series of motion nodes, and in each node the motion is constant. Each node corresponds to the acquisition of one k-space line or one group of k-space lines. The starting point for this trajectory is zero motion corresponding to one group of k-space lines, while the motions in other groups of k-space lines are corrected by assessing the image quality, named entropy focus criterion. The motion estimation is modified with trial-and-error iteratively to obtain the best quality image.
The entropy of an image $E$ is defined as:

$$E = - \sum_{i=1}^{N} \frac{|p_i|}{p_{\text{total}}} \ln \left( \frac{|p_i|}{p_{\text{total}}} \right),$$  \hspace{1cm} (2.5)$$

where $p_i$ is the pixel value of the image, and $N$ is the total number of pixels of the image. $p_{\text{total}}$ is given by:

$$p_{\text{total}} = \sqrt{\sum_{i=1}^{N} p_i^2}. \hspace{1cm} (2.6)$$

Entropy minimization prefers high contrast and increasing the number of dark pixels, while motion tends to do the opposite by creating ghosts and blurring images in the dark regions. For an image, when all the image energy is in one pixel and the other pixel values are 0, $E$ equals 0, which is the minimum of (2.5). But when all the pixel values in a $256 \times 256$ images are equal, the image entropy $E$ is around 1420, which is the maximum of (2.5) for that image size.

The optimization strategy is similar to a multi-resolution search scheme. The readouts within the acquisition are grouped into several segments, and each segment includes the same number of non-overlapping MRI k-space lines. In order to illustrate this clearly, we take 256 readouts as an example. The k-space is separated into 16 segments, each having a length of 16 k-space lines. The 16 segments are grouped into two parts—lower part and upper part. Each part consists of 8 segments. The lower part is from the scan center to the end of the scan, and the upper part is from the scan center to the start of the scan. The first segment of the lower part consists of 16 k-space lines from the scan center (low frequency) towards the end of the scan (high frequency). This segment has trial translations in both the phase encoding and frequency encoding directions $\{ -N, -(N-1), -1, 0, 1, (N-1), N \}$ ($N$ is the searching range), with a total of $(2N+1) \times (2N+1)$ combinations for this segment. For each combination, preserving other segments unchanged, the image entropy is calculated by (2.5). The lowest entropy of the combination is accepted, and the motion of the first segment of the lower part is corrected corresponding to the combination. The subsequent
segments of the lower part are considered similarly in turn. The segments of the upper part are calculated in the same way. This process is repeated, and the segment length decreases by two k-space lines in each iteration. In this way the gross motions are determined first and gradually refined.

This method has high computational load due to using a trial-and-error approach. In addition, the optimization strategy makes the search more likely to be trapped in local minimum.

2.5.2 blind retrospective motion correction of MR images

To reduce computational load for 3D rigid motion, an analytical gradient-based motion correction (GradMC) algorithm was proposed by Loktyushin et al. [59]. A parameterized 3D motion-corrupted k-space model was proposed. Entropy of the gradient of the image is used as the focus criterion in this method. The motion assumption is similar to [8] but in 3D space. Quasi-Newton and Broyden-Fletcher-Goldfarb-Shanno is used to optimize the cost function iteratively.

Entropy of gradient of an image is defined as:

\[ \phi(I) = H(D_x I) + H(D_y I), \]  

(2.7)

where \( D_x \) and \( D_y \) are the difference matrices in horizontal and vertical directions and \( H() \) is defined as:

\[ H(p) = -v^T \ln v, v = \sqrt{p \odot p^*} \]

(2.8)

where \( \odot \) is the point-wise product of two vectors, \( p \) is the pixel vector of the image \( I \), * represents the complex conjugate, and \( H \) represents complex conjugate transpose.

Coarse to fine optimization strategy is used. The coarse image formed by only lower frequency k-space data is corrected first, and then a finer image is formed by adding a higher-frequency segment. All the k-space lines that formed the finer image are corrected based
on the focus criterion. The process iteratively covers all the k-space lines and corrects the whole k-space data, which is shown in Figure 2.7.

![Figure 2.7: Optimization strategy of Blind Retrospective Motion Correction of MR Images. The k-space data is processed from center (low frequency part) to the whole k-space to optimize motion parameters using coarse to fine optimization. (Figure 2.7 is from [59].)](image)

Since the algorithm performs line-by-line 6-parameter correction, the calculation load is very high. Hence, a highly parallel implemented GPU version gradient-based motion correction optimization strategy is used. A similar idea was used to correct nonrigid motion by correcting each rigid motion patch [60].

### 2.5.3 nonrigid motion correction in 3D using autofocusing with localized linear translations

An autofocusing motion correction method based on “Butterfly” navigators to reduce calculation load and correct nonrigid motion was introduced by Cheng et al. [19]. The different motion on each small patch of image is approximated by a translation. “Butterfly” navigators provide possible translational motion information to reduce the search range. Localized entropy of the gradient of the image is used as the focus criterion. Based on this localized autofocusing framework, a new motion correction method combined with other sample patterns and compressed sensing was proposed [20].
2.6 Artificial neural network methods

Artificial neural networks (ANNs) are widely used in machine learning. In general, an ANN is made of a collection of connected computational units, called artificial neurons, which simulates the biological neurons. Artificial neurons are arranged in layers. Different network structures are formed by different number, arrangements and connections of layers, or different orders of functional layers. A loss function is a measure function to compare the real output (the labeled output from the training set) with network output. An ANN is trained to minimize a loss function using a labeled training set. The details of ANNs will be discussed in the Chapter 6.

Neural networks are widely used in the MRI field: image reconstruction [105, 95, 82], image denoising [87, 13, 12], image super-resolution [109, 57, 17], and image segmentation [18, 42, 21]. The history of using neural networks to correct MRI motion is relatively short. In general, these methods require pairs of images as the training set: motionless images and corresponding motion-corrupted images but using different network structures and loss functions. Loktyushin et al. [61] proposed a motion correction method using convolutional neural networks. Duffy et al. [24] used a modified HighRes3dNet (HR3DNet) [56] to correct MRI head motion. Pawar et al. [75] implemented an encoder-decoder convolutional neural network to restore motion-blurred images. Armanious et al. [7] proposed a novel framework called MedGAN based on GANs for medical image translation.

2.7 Advantages and limitations of PMC and RMC

In this section, we discuss the advantages and limitations of PMC and RMC separately and at the end discuss one hybrid motion correction method.

2.7.1 advantages of PMC

data consistency and uniform sampling
In PMC, the FOV (field of view) follows the object tightly, so the object cannot move out of the imaging volume in normal situations. Uniform sampling can be kept in PMC; on the other hand, RMC cannot perform uniform sampling, especially in the case of large rotation.

*through-plane motion*

Spin-history effects occur when the object moves in and out of the imaging slice [33]. Through-plane motion can disrupt the steady state in one imaging slice, leading to signal fluctuations in conventional imaging. PMC can correct the through-plane motion to reduce the spin-history effects.

*real-time imaging*

Imaging time is an important factor in research and clinical applications. Some modern PMC can provide real-time imaging.

*calculation load*

Calculation load is relatively low for PMC, and there is no additional calculation performed for the image reconstruction.

### 2.7.2 limitations of PMC

*tracking motion error effect*

The performance of PMC is determined by the acquisition of reliable and accurate tracking data. Inaccurate pose data or noisy data can corrupt the image information. Cross-calibration errors may be fatal for optical PMC.

*additional hardware*

For optical and field detection methods, additional hardware is needed to track the motion, which makes the whole system complicated and expensive.

*sequence adjustments*
For the MR-navigator methods, some MRI sequences have to be changed to add the navigators, which may increase the scan time and lead to complicated sequence modification. In some cases, navigators are not compatible with sequences.

*marker attachment*

Markers have to be attached to the object securely to track the motion, which results in inconvenience and discomfort for the subject. If markers move, erroneous updates will be applied, which may destroy image quality.

### 2.7.3 advantages of RMC

*simple system*

For autofocusing methods, no additional hardware, no gradient update, no modification of MRI sequences and no markers are needed. For neural network methods, there are no specific requirements for the software and hardware of scanners. A basic MR system is enough for RMC. Accounting for the equipment cost and operation complexity, RMC is suitable for clinical use.

*real-time motion correction*

Compared with other RMC methods, we propose a new motion correction method based on fast robust correlation that can provide real-time motion correction. For artificial neural network methods, if the training time is ignored, the motion correction can be accomplished in real time.

*no preparation required*

Because no special sequences or hardware is required, no special preparation is required prior to or during acquisition. A decision about performing motion correction can be made after the data is acquired.

### 2.7.4 limitations of RMC

*through-plane motion*
2D RMC cannot correct through-plane motion, which may lead to motion correction errors and induce image artifacts.

*non-realistic images*

Non-realistic motion-corrected images may be generated by artificial neural networks methods. It is hard to eliminate this phenomenon, which is determined by the nature of the artificial neural network.

*nonuniform samples*

According to the property of Fourier transforms, correction of large angle rotations may result in nonuniform sampling and local Nyquist violations, which is shown in Figure 2.8.

![Figure 2.8: (a) Sample locations after RMC correction for rotation object. (b) Sample locations after PMC for rotation object.](image)

*calculation load*

The calculation load is relatively high for previous RMC methods. The most common way to reconstruct images is to iteratively update motion parameters.

*local minima*

RMC are easily trapped in local minima, due to the vast motion search space and imperfect optimization strategies.
A combination of PMC and RMC is possible and some of the limitations can be addressed partially [14, 65]. Aksoy proposed a motion correction method to combine prospective optical-based motion correction with removal of residual motion artifacts by entropy-based retrospective autofoocusing [2]. In general, scanner-camera cross-calibration is the key factor with optical prospective motion correction, because a small error in the cross-calibration can result in wrong position adjustment for the scanner and induce image artifacts. However, for this hybrid method, the requirement of cross-calibration accuracy is decreased, and high-quality images may still be obtained, because the autofocusing motion correction method is applied after data is acquired. The details of this method will not be discussed here.
Chapter 3
Image Registration

Image registration is closely related to some motion correction methods and our proposed methods. In this chapter, some basic image registration concepts are first discussed. Image registration is defined as a process by which different sets of data are transformed into the same coordinate system. This process sets one image as the target (also called the fixed image, or the reference image), and implements a transformation model to the other image (also called the moving image, the floating image, or the source image) to match the target image.

Image registration is a basic technology in image processing. The applications can be generally classified into three fields [112].

*different times.* Images of the same object are acquired at different times, and the purpose of registration is to find the changes of the object, such as motion tracking, landscape changes and tumor evolution.

*different views.* Images are obtained from different viewpoints at the same time, and the purpose of registration is to get a bigger view of images, such as remote sensing and 3D image reconstruction.

*different sensors.* Images are acquired from different sensors for the same object, and the purpose of registration is to catch and fuse more details of the same object, such as remote sensing, multiple sensor fusion, MR-CT, and MR-SPECT fusion.

Figure 3.1, Figure 3.2 and Figure 3.3 give examples of different times, different views, and different sensors image registration.
Figure 3.1: Different times brain images registration to detect a lesion. (This image is from the link https://thebestphotos.eu/detection-of-focal-longitudinal-changes-in-the-brain-by.html)

Figure 3.2: Different views aerial images registration to extend the view. (This image is from MathWorks website Image Registration section, and the link is https://www.mathworks.com/discovery/image-registration.html)

Figure 3.3: Different sensors brain images registration to fuse images. (This image is from UCF Medical Image Computing (CAP 5937) slices (Spring 2016), and the link is http://www.cs.ucf.edu/~bagci/teaching/mic16/lec15.pdf)
The image registration methods can be classified into two groups: area-based registration and feature-based registration. The process of both registration methodologies show in Figure 3.4 and Figure 3.5.

In general, a registration algorithm, whether area-based or feature-based, can be decomposed into three parts:

- **transformation models.** The way that a source image can be changed to match the target image.

- **matching criteria.** The similarity measures of two images to be matched.

- **optimization methods.** Algorithms to maximize or minimize the matching criteria.
For the feature-based image registration, one more step—feature detection—is needed. Considering the relation to the proposed methods, we only discuss area-based image registration here. This chapter is organized as follows: 3.1 transformation models, 3.2 matching criteria, and 3.3 optimization methods.

### 3.1 Transformation models

A transformation model defines how one image can be deformed to align with the other image, which is an important step for registration. The choice of transformation model depends on the image registration application and the nature of the image data, which influences the rest of the registration steps. The transformation model has two functions. First, it improves the similarity of the images by defining the source image move relative to the target image. Second, it defines the source image interpolation if necessary. In general, transformation models can be classified into three primary groups: rigid, affine and non-rigid models.

The rigid transformation model is a basic and widely used model in image registration. Rigid transformation preserves the Euclidean distance for each point pair in a Euclidean space, which is also called a Euclidean transformation. In two-dimensional space, the model is always defined by three parameters: two translation parameters and one rotation parameter, which is described as three degrees of freedom. These parameters are applied to a vector locating a point in the source image to find its location in the target image. Different orders of rotation and translation operations may have different transformation model representations [31, 37]. For rotation before translation, the transformation model is given by:

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} = \begin{bmatrix}
    \cos \theta & \sin \theta & t_x \\
    -\sin \theta & \cos \theta & t_y \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix},
\] (3.1)
where \( t_x \) and \( t_y \) specify the translations along the \( x \) and \( y \) axes, and \( \theta \) is the angle of rotation at the center of the image.

The corresponding model for translations before rotation is given by:

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & \sin \theta & t_x \cos \theta + t_y \sin \theta \\
  -\sin \theta & \cos \theta & -t_x \sin \theta + t_y \cos \theta \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}.
\] (3.2)

In three-dimensional space, the model includes six parameters: three translation parameters and three rotation parameters. Here we just show the representation for rotations before translations:

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  \cos \omega \cos \phi & \cos \theta \sin \phi + \cos \phi \sin \omega \sin \theta & \sin \phi \sin \theta - \cos \phi \sin \omega \cos \theta & t_x \\
  -\cos \omega \sin \phi & -\sin \theta \sin \omega \sin \phi + \cos \theta \cos \phi & \sin \theta \cos \phi + \sin \phi \sin \omega \cos \theta & t_y \\
  \sin \omega & -\sin \theta \cos \omega & \cos \theta \cos \omega & t_z \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}.
\] (3.3)

where \( t_x \), \( t_y \) and \( t_z \) specify the translations along the \( x \), \( y \) and \( z \) axes, and \( \theta \), \( \omega \) and \( \phi \) are the angles rotation around the \( x \), \( y \) and \( z \) axes.

A rigid transformation model is mainly used in two ways. One is to model rigid structure motion, such as head MRI. The other is for pre-registration of non-rigid structures, that is then followed by more complicated non-rigid registration based on the pre-registration.

An affine transformation model can permit translation, rotation, scaling and shearing deformations [49]. This model can align images with perspective distortions. Figure 3.6 shows this model. The transformed coordinates are given by:
\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} & t_x \\
  a_{21} & a_{22} & a_{23} & t_y \\
  a_{31} & a_{32} & a_{33} & t_z \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix},
\]  

(3.4)

where \(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}\) and \(a_{33}\) specify rotation, scaling and shearing deformations, and \(t_x, t_y\) and \(t_z\) specify the translations.

Figure 3.6: Affine transformation model, the parallel lines keep parallel after an affine transformation.

Non-rigid registration is a popular topic in recent research. One commonly used and important non-rigid registration method is a group of splines technique [89, 53, 54]. In splines registration methods, corresponding control points are defined in both target and source images to calculate distance between these control points by spline function. Thin-plate splines [25] and B-splines [103] are the most widely used among the spline methods. Thin-plate splines registration is a global method, which means each control point in the thin plate has a global influence on the transformation model. If the position of one control point changes, the transformation model will change, and the whole transformed image will change. By contrast, B-splines is a local registration method, which is defined in the vicinity of each control point. So one control-point position change will only affect transformation in the neighborhood of the point, not the whole transformed image. Hence, it is more robust. Figures 3.7 and 3.8 show the thin-plate splines and B-splines registration.
3.2 Matching criteria

Area-based matching criteria align images by evaluating the whole image domain pixel values. This strategy is widely used when image features are not salient or the features are hard to detect, such as some kinds of medical image registration. Since the whole data is used, the registration is more robust, and error pixels do not have much effect on the final transformation model. However, computational expense is increased when the size of the image is large.

The area-based matching criteria include two similarity measures: intensity-based and frequency-based.
Intensity-based similarity measures can be divided into three groups: based on intensity differences, based on intensity cross-correlation and based on information theoretic approaches [22, 44].

Intensity difference methods usually use the sum of squared or absolute differences (SSD or SAD) as a matching criterion. In order to use the intensity differences similarity measure, the same structures of the target and source images should have similar intensity values. Thus, the minimum of SSD or SAD means two images matched. SSD and SAD are given by:

\[
SSD = \frac{1}{N} \sum_{x \in \Omega_{T,S}} \left[ T(x) - S(t(x)) \right]^2, \tag{3.5}
\]

\[
SAD = \frac{1}{N} \sum_{x \in \Omega_{T,S}} \left| T(x) - S(t(x)) \right|, \tag{3.6}
\]

where \( T(x) \) is the intensity of voxel \( x \) in the target image and \( S(t(x)) \) is the intensity of the corresponding voxel in the source image through the estimation of the transformation function \( t(x) \), \( \Omega_{T,S} \) is the overlap region of the target and the source images and \( N \) is the number of voxels in this region.

The cross-correlation (CC) method has also been used as an image similarity measure. The assumption behind CC is that there is a linear intensity relation between the two images. Compared with SSD, the maximum of CC means two images are correctly aligned. CC is given by:

\[
CC = \frac{\sum_{x \in \Omega_{T,S}} \left( T(x) - \bar{T} \right) \left( S(t(x)) - \bar{S} \right)}{\sqrt{\sum_{x \in \Omega_{T,S}} \left( T(x) - \bar{T} \right)^2} \sqrt{\sum_{x \in \Omega_{T,S}} \left( S(t(x)) - \bar{S} \right)^2}}, \tag{3.7}
\]

where \( \bar{T} \) is the mean voxel intensity of the target image and \( \bar{S} \) is the mean voxel intensity of the source image.

SSD and CC are used for mono-modal registration, but for multi-modal, registration information theoretic approaches are the most popular.
Joint distribution of the Shannon entropy provides one way to measure the alignment [78], which is given by:

\[ H(T, S) = - \sum_{t,s} p_{T,S}(t, s) \log p_{T,S}(t, s), \]

where \( p_{T,S}(t, s) \) is a joint probability density function of target and source images in the overlap region. The entropy is low when the distribution has concentric peaks, and it is high when the distribution is more uniform. So the strategy is to minimize the joint entropy to match the images. The joint probability density function is generated from a joint histogram of the target and source images.

When joint distribution of the Shannon entropy is used alone, misregistration can occur. For example, when two images only have one piece of background overlaps, the joint histogram has only one sharp peak, and the joint entropy has a low value that leads to an uncorrected registration. Mutual information (MI) [66, 99] is an effort to solve this problem. MI measures the amount of information in the overlapping volume by the marginal entropy of the individual image being registered together with the joint entropy, which is given by:

\[ MI(T, S) = H(T) + H(S) - H(T, S), \]

where \( H(T) \) and \( H(S) \) are the marginal entropies of each image. Eq 3.10 and Eq 3.11 give the overlap region marginal entropy of each image.

\[ H(T) = - \sum_{t} p_{T}(t) \log p_{T}(t) \quad \forall T(x) = t | x \in \Omega_{T,S}, \]

\[ H(S) = - \sum_{s} p_{S}(s) \log p_{S}(s) \quad \forall S(x) = s | x \in \Omega_{T,S}. \]

MI can be further expressed as:
\[ MI(T, S) = H(T) + H(S) - H(T, S) = \sum_{t,s} p_{T,S}(t,s) \log \frac{p_{T,S}(t,s)}{p_{T}(s)p_{S}(t)}. \tag{3.12} \]

For the previous example, the marginal entropies will have low values when the overlapping region only includes background, and they will have high values when the overlapping region includes sharp structures. The marginal entropy terms provide a tradeoff with the joint entropy.

All three previously discussed similarity measures are in the spatial domain, but there is also a group of methods whose similarity measures are defined in the Fourier domain [46, 32]. If data is processed more easily in the frequency domain or an acceleration of the computation speed is needed, Fourier-domain methods can be a more powerful tool. This group of methods is a natural fit with MRI, because the image data is acquired by sampling in the Fourier domain (k-space). The phase correlation method (PCM) is the most common method in this group. According to the Fourier shift property, a linear phase difference in the Fourier transform of two images comes from a translation of the images, which is given by

\[ F_S(k,l) = F_T(k,l) \exp\{-j(kt_x + lt_y)\}, \tag{3.13} \]

where \(F_S\) and \(F_T\) are the Fourier transform of source and target images. \((k,l)\) are the Fourier domain coordinates.

The normalized cross-power spectrum is used to calculate phase correlation matrix, which is given by:

\[ Q(k,l) = \frac{F_T(k,l)F_S^*(k,l)}{|F_S(k,l)F_S^*(k,l)|} = \exp\{-j(kt_x + lt_y)\}. \tag{3.14} \]

PCM is robust against frequency-dependent noise and time-varying illumination disturbances. However, PCM is used only for registration of translations of images.
3.3 Optimization methods

Optimization is required to find the transformation to best align two images, which means finding the minimization or maximization of similarity measures. A good optimizer should be reliable and quick to find possible solutions. Different registration problems require different optimizers.

The similarity measure of registration can be accounted as an $n$-dimensional function, where $n$ is the number of parameters of the transformation model. In order to find the $n$ parameters to minimize or maximize this function, iterative optimization algorithms are the most common strategy.

In general, optimizing $n$ parameters follows an update below:

$$s_{i+1} = s_i + \lambda_i g_i(s_i), \quad (3.15)$$

where $s$ is the vector of $n$ parameters of a similarity measure function, $\lambda$ is the step length, $g$ is the search direction, and $i$ defines the number of the iteration.

Typical methods to update $n$ parameters include: gradient descent (GD), conjugate gradient (CG), Quasi-Newton (QN), Levenberg - Marquardt (LM) and Gauss-Newton (GN) methods [89]. The main difference among these algorithms is how to find the next step search direction, namely $g$. The equations of the above algorithms are shown in Table 3.1.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Search directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>$g = -\nabla s(s)$</td>
</tr>
<tr>
<td>CG</td>
<td>$g = -\nabla^2 s(s_i) + \beta_i g_{i-1}$</td>
</tr>
<tr>
<td>QN</td>
<td>$g = -\hat{H}^{-1}(s) \nabla s(s)$</td>
</tr>
<tr>
<td>LM</td>
<td>$g = -(\hat{H}^{-1}(s) + \alpha I) \nabla s(s)$</td>
</tr>
<tr>
<td>GN</td>
<td>$g = -(J^T(s)J(s))^{-1} \nabla s(s)$</td>
</tr>
</tbody>
</table>
A multi-resolution strategy is generally used in these iterative algorithms, which is also known as a coarse-to-fine strategy [72]. First, both the target and the source images are down-sampled. Then these two down-sampled images are aligned. Second, the transformation parameters from the last step are used as the initial guess for higher-resolution images, which come from the original images with a smaller downsampling rate. So the registration is from lower- to higher-resolution images until the original resolution is reached. In each step, the parameters found in the previous step are used to obtain the initial guess of the current step. This strategy can avoid the local minimum problem and increase computational speed.
Chapter 4

An Efficient and Robust Autofocusing MR Image Motion Correction Method using Global Motion Estimation

Autofocusing motion correction methods are one kind of retrospective MRI motion correction method. These methods assume a parameterized motion model and estimate the model parameters by iterative optimization of an image quality metric, called a focus criterion. Motion is corrected after the raw k-space data are modified according to the optimized model parameters. Implementing these methods does not require additional hardware, gradient update, modification of MRI sequence or markers attached on subjects. However, due to the vast size of the search space, autofocusing methods always have two challenges: high calculation load and local minima.

In order to address the two challenges, a new autofocusing motion correction algorithm is proposed in the chapter. We propose to use multiple linear-motion initializations and joint refinement of a global model to decrease and constrain the search space. In the first step, k-space is divided into several segments based on acquisition order. Linear motion is assumed and searched in each segment to get initial motion parameters. In the second step, several control points are chosen on the piecewise linear initial approximation, and then a piecewise cubic Hermite interpolation polynomial is fitted from the control points to obtain smooth motion curves. The motion curves are refined by optimizing a focus criterion. These strategies make the proposed algorithm efficient and robust. Different focus criteria are compared under the proposed method. To further improve computational efficiency, golden-section search is used to estimate rotation, and two map data structures are applied to store calculated data. Simulations and experimental results demonstrate that the proposed algorithm can effectively and efficiently correct motion in MR images.
This chapter is organized as follows: 4.1 motion model and acquisition assumption, 4.2 the problem of previous search strategies, 4.3 focus criteria, 4.4 two steps of the proposed method, 4.5 golden-section rotation search, 4.6 map data structure, 4.7 rotation and translations interpolation methods, 4.8 simulations, 4.9 experiments, and 4.10 discussion.

### 4.1 Motion model and acquisition assumption

In this chapter, the motion is assumed to be two-directional translations and one rotation in a 2D plane. It is assumed that the $x$ axis corresponds to the frequency encoding direction and the $y$ axis corresponds to the phase encoding direction. The acquisition time gap of different $k_{\text{phase}}$ lines is much longer than the duration of the frequency encoding (acquisition of one $k_{\text{phase}}$ line). So the motion during the frequency encoding (one single $k_{\text{phase}}$ line) is much smaller than the inter-line (different $k_{\text{phase}}$ lines) motion, which means intra-line motion can be assumed negligible. The basic unit for motion correction in this chapter is chosen to be a single $k_{\text{phase}}$ line, which means we only consider motion between different $k_{\text{phase}}$ lines. Within each $k_{\text{phase}}$ line, the object is modeled as stationary.

The motion parameters $\Delta x$, $\Delta y$ and $\Delta \theta$ corresponding to $x$ and $y$ translations and rotation angle are a function of $k_{\text{phase}}$. Therefore, the motion correction is essential to estimate a total of $N_{\text{phase}}$ 3D vectors $M$, where $N_{\text{phase}}$ is the number of $k_{\text{phase}}$ lines. The motion parameters can be given by:

$$M(k_{\text{phase}}) = (\Delta x(k_{\text{phase}}), \Delta y(k_{\text{phase}}), \Delta \theta(k_{\text{phase}})),$$  \hspace{1cm} (4.1)

This motion model is shown in Figure 4.1.
Figure 4.1: The motion model of the proposed method. In k-space, each dash line is one $k_{\text{phase}}$ line. Different $k_{\text{phase}}$ lines have their own motion parameters $(\Delta x, \Delta y, \Delta r)$. Red, green and blue are three $k_{\text{phase}}$ lines with three different sets of motion parameters.

4.2 Previous search strategies

In general, the previous search strategies can be classified into two groups. One is to search one segment of k-space data, which includes several neighboring k-space lines, while keeping the remaining k-space lines unchanged. Then a different segment is searched until all segments covering the whole k-space are corrected. After each iteration, the size of the segments is reduced to refine motion. The other strategy is to preserve the central k-space part as the reference and then detect and correct the motion of neighboring peripheral k-space segments. The motion-corrected peripheral k-space segments are combined with the central k-space part to form a new reference, and further peripheral k-space segments are corrected until the whole k-space is covered.

We found that neighboring lines in k-space have a strong relationship, and the motion detection of one segment may not correspond to the average motion of all the lines in the segment when the segment size is smaller than a certain number. Sometimes the motion estimate of a segment matches the motion of the nearest line to the reference segment.
rather than matching the average motion of the k-space segment to the reference. Due to the strong impact of the neighboring line, the motion of the whole segment may not be corrected properly, which may result in the cost function being trapped in local minima, which is shown in Figure 4.2. In order to overcome this problem and take advantage of the strong relationship between the neighboring lines, a new search strategy based on global motion estimation is proposed in the next section.

![Figure 4.2](image)

Figure 4.2: The strong relationship between neighboring lines results in motion estimation error. Segment 1 and segment 2 are two neighboring segments in k-space and the green solid line gives the real motion for each k-space line of the segment 1 and the segment 2. Due to the strong bond between the red and blue thick solid lines, the estimated motion of the segment 1 is dominated by the red thick solid line, which results in the errors for other lines in the segment 1.

### 4.3 Focus criteria

The focus criterion is important for image evaluation in autofocus methods. Considering accuracy and efficiency, an ideal criterion should satisfy three conditions: first, the criterion value can measure the image quality correctly; secondly, the extremum corresponds to the best image quality. Thirdly, the criterion should be easy calculated. Many focus
criteria have been proposed [68] that general meet the above conditions. We cannot pro-
vide theoretical proof of which is the optimal criterion under the proposed method, but we
can draw a well-grounded conclusion based on empirical performance. Image entropy and
different kinds of entropy of the gradient of image are the most popular criteria for auto-
ofocusing methods [68, 8, 59, 19]. Based on [68], the entropy of one-directional gradient
along the phase-encoding direction is the best criterion. But this conclusion relies on certain
evaluation rules, which may not be suitable for the proposed motion correction method.
To our knowledge, the same focus criterion may perform differently under various motion
correction methods. So in this chapter, the image entropy (Eq 4.2), the entropy of the one-
directional gradient of image along the phase-encoding direction (Eq 4.4), the entropy of the
two-directional gradients of image (Eq 4.5), the total variation of an image (Eq 4.7) and the
sum of the absolute values (Eq 4.9) will be compared under the proposed motion correction
method.

The entropy of an image is defined as:

\[ c_1 = - \sum_{i=1}^{N} \frac{|p_i|}{p_{\text{total}}} \ln \left( \frac{|p_i|}{p_{\text{total}}} \right), \]  

(4.2)

where \( p_i \) is the pixel value of the image, and \( N \) is the total number of pixels of the image. \( p_{\text{total}} \) is given by:

\[ p_{\text{total}} = \sqrt{\sum_{i=1}^{N} p_i^2}. \]  

(4.3)

The entropy of the one-directional gradient of an image is defined as:

\[ c_2 = H(D_y I). \]  

(4.4)

The entropy of the two-directional gradients of an image is defined as:

\[ c_3 = H(D_x I) + H(D_y I), \]  

(4.5)
where $D_x$ and $D_y$ are the difference matrices in the horizontal and vertical directions, $I$ is the input image, and $H()$ is defined as:

$$H(p) = -v^T \ln v, \quad v = \sqrt{p \odot p^*} / p^H p,$$

where $\odot$ is the point-wise product of two vectors, $p$ is the pixel vector of the image $I$, $*$ represents the complex conjugate, and $H$ represents complex conjugate transpose.

The total variation of an image is defined as:

$$c_4 = S(D_x I) + S(D_y I),$$

where $S()$ is defined as:

$$S(p) = \sum_{i=1}^{N} |p_i|,$$

where $p_i$ is the pixel value of the image.

The sum of the absolute values of an image is defined as:

$$c_5 = \sum_{i=1}^{N} |p_i|.$$

4.4 Proposed method

In order to decrease computational complexity and avoid being trapped in local minima, the proposed method involves two steps: piecewise linear initial approximation and joint refinement of a global model to perform global estimation.

piecewise linear initial approximation

K-space is divided into several segments based on acquisition order. Linear (uniform speed) motion (two translations and one rotation) is assumed in each segment to get initial motion parameters, which are the slopes (the speed) of the linear motions. In order to obtain the slopes of these linear motions, searching is done within possible slopes to optimize the
criterion. A linear motion approximation is reasonable, since the acquisition time in each segment is relatively short. Subject motion velocity is expected to be in a small range within a small time frame, which makes the search space small and improves search efficiency. The process is shown in Figure 4.3. Due to the three parameters of the motion model, the search space consists of three levels of loops. In order to further improve search efficiency, golden-section search is introduced to reduce search times, which will be discussed in the later section.
Figure 4.3: The process of piecewise linear initial approximation. There are three motion curves with the $x$ axis representing $k_{	ext{phase}}$ lines and the $y$ axis representing pixels for $x$, $y$ translations and rotation degree for rotation. The whole $k$-space is divided into three segments with a certain number of $k_{	ext{phase}}$ lines in each segment. In Segment 1, the linear estimations of the three motions are chosen from 729 ($9 \times 9 \times 9$, in each motion curve, 9 slopes (blue dashed line) are given) different motion combinations, which can minimize the criterion of the image formed by Segment 1, marked as red lines. In Segment 2, the linear estimations of the three motions are chosen from 729 different motion combinations all starting from the linear estimation of Segment 1, which can minimize the criterion of the image formed by Segment 1 and Segment 2, marked as green lines. The process is repeated for Segment 3.

**joint refinement of global model**

To approximate exact motion, joint refinement of global refinement is implemented on the piecewise linear initial approximation. Several equally spaced control points are chosen
on the piecewise linear initial approximation. Based on the assumption that subject motion is smooth, a piecewise cubic Hermite interpolating polynomial using MATLAB function `pchip` is fitted from the control points to obtain smooth motion curves. To further refine motion curves, the MATLAB function `fminsearch` is used to globally optimize the criterion by refining the cubic Hermite interpolation curves iteratively. In each iteration, one more control point is added from the previous refined motion curve to provide more degrees of freedom for the motion curve until the desired model precision is accomplished. In each iteration, three motion curves (two translational curves and one rotation curve) are considered as a system to optimize the criterion. The Figure 4.4 shows the above process.
Figure 4.4: The process of joint refinement of global model. In the first row, four equally spaced control points (black cross) are chosen on the piecewise linear approximation (red linear lines) for each motion parameter, and the globally criterion optimization is accomplished by refining three piecewise cubic Hermite interpolating polynomials (three green dashed curves) fitted from the four control points. One more control point is added from the previous refined motion curves (red curves in the second row) to get a refined curves (green dashed curves in the second row). The process is repeated until the desired model precision is accomplished (three red curves in the third row).

To handle the nonuniform sampling in k-space after correction of large angle rotations (Figure 2.8), image rotation correction is performed in the spatial domain. After the image formed by one k-space segment is corrected in the spatial domain, the image is transformed
back to the frequency domain, and then the same segment is acquired by resampling the frequency domain. The whole process is shown in Figure 4.5.

![Image of rotation correction process]

Figure 4.5: The process of rotation correction. The frequency domain images are logarithmically scaled.

### 4.5 Golden-section search

To make rotation search more efficient, golden-section search is implemented. Here the golden-section search method is discussed in detail first.

Golden-section search is used to find the extremum of a strictly unimodal function by reducing the value range successively [101]. The function values at three points are always maintained, and their distances form a golden ratio. This algorithm, which was first introduced by Kiefer [52], can narrow the range where an extremum exists efficiently. The function values at three points are compared to further narrow the search range.

Figure 4.6 illustrates one step of a golden-section algorithm to find a minimum. $f_1$, $f_2$, and $f_3$ are known values, and $f_2$ is smaller than $f_1$ and $f_3$. The minimum exists in $[x_1, x_3]$. 

47
$x_4$ is used to further narrow the search range. If $f_{4a}$ is the function value of $x_4$, the minimum exists in $[x_1, x_4]$, and $x_1$, $x_2$ and $x_4$ are new triplet points needed to be maintained. If $f_{4b}$ is the function value of $x_4$, the minimum exists in the $[x_2, x_3]$, and $x_2$, $x_4$ and $x_3$ are new triplet points need to be maintained. Search range is narrowed at each step.

The selection of the position of $x_4$ is important. The new search range is either $[x_1, x_4]$ or $[x_2, x_3]$. In order to reduce iterative times, the length of $[x_1, x_4]$ and the length of $[x_2, x_3]$ should be the same: $b = a + c$. So $x_4 = x_1 + (x_3 - x_2)$. Since $x_2$ should have the same proportion of space as the new triples points $x_1$, $x_2$, $x_4$ and $x_2$, $x_4$, $x_3$, two equations are derived below:

If $x_1$, $x_2$, $x_4$ are the new triplet points:

$$
\frac{c}{a} = \frac{a}{b}. \tag{4.10}
$$

If $x_2$, $x_4$, $x_3$ are the new triplet points:

$$
\frac{c}{b - c} = \frac{a}{b}. \tag{4.11}
$$

From Eq 4.10 and Eq 4.11, we can show that $b/a$ is equal to the golden ratio.

Figure 4.6: One step of golden-section search to find the minimum. (Figure 4.6 is from [101].)
We cannot prove that the criterion values in the rotation search range satisfy a strictly unimodal function condition, but all numerical simulations show they can be treated as a strictly unimodal function.

Golden-section search is used to reduce rotation search times in the step of piecewise linear initial approximation. For each trial of translations, the rotation is chosen by golden-section search to minimize a criterion.

4.6 Map data structure

To eliminate repeated computation, two map data structures are used to improve efficiency. A map is a data structure which is used for fast lookup or data searching. A map uses key-value pairs to store and quickly search data. Every key is unique and maps to a value. In our method, one map is used to store the rotated k-space data with rotation angle as the key, since the image rotation is time consuming and many repeated calculations are potentially needed. The key-value pairs of the map are illustrated in Figure 4.7. The other map is used to keep the criterion for previously evaluated correction parameters in the golden-section search of rotation. Golden-section search decreases rotation search times, and the map data structure saves the calculated results for further use. The processing time is cut in half by using golden-section rotation search and the map data structure.
Figure 4.7: The map is used to store the rotated k-space data with rotation angle as the key. When the rotation angle $-3.38^\circ$ is searched, the corresponding rotated k-space matrix $F_{6M \times N}$ is returned.

4.7 Interpolation methods

Two interpolation methods—rotation interpolation and translation interpolation—are used in the proposed method. In order to preserve the image quality of rotated images, zeropadding in the frequency domain to interpolate the image is used before image rotation, which is shown in Figure 4.8. This strategy effectively reduces the decay of high frequencies of rotated images arising from nonideal interpolation. But interpolation in the bigger image increases calculation time greatly. A trade-off is made between image quality and processing time. Figure 4.9 shows the error of the image with zeropadding rotation interpolation is much smaller than the image with no zeropadding rotation interpolation. In order to obtain subpixel translation accuracy, translation interpolation is achieved in the frequency domain, according to the properties of the Fourier transform:

$$
\begin{align*}
\begin{cases}
  f(x, y) & \iff F(u, v), \\
  f(x - a, y - b) & \iff e^{-j2\pi(au+bv)}F(u, v).
\end{cases}
\end{align*}
$$

$$\text{ (4.12)}$$
Figure 4.8: Rotation interpolation. The frequency domain images are logarithmically scaled. The length and width of the zeropadded images are twice the original image.

Figure 4.9: The difference images comparison. The same image is used as the reference images but different floating image. For (a), the floating image is generated by Figure 4.8. For (b), the floating image is generated by rotating the reference image directly. In order to acquire the same orientation with the reference image, both floating images rotate an angle clockwise and then rotate the same angle counterclockwise. (In order to rotate clockwise and counterclockwise, for (a), the padding zero image is used to rotate and then downsampled.) Both difference images are shown at the same scaling. The dominant error occurs around the edge in (a), while the image content error is the main error in (b) with a bigger margin.
4.8 Simulations

In order to validate the proposed method and compare the five criteria discussed above using the proposed method, simulations were conducted in this section. Three kinds of motion—piecewise linear, parabolic, and sinusoidal—were implemented to simulate two-translational and rotational motion in the 2D plane on a reference image. In order to better model real motion, the parameters of motion were chosen randomly within certain ranges. Fig 4.10 shows one example of each of the three motions.

![Figure 4.10: Three different motions.](image)

Figure 4.10: Three different motions. The number of function values (160) corresponds to the number of \(k_{\text{phase}}\) lines. Horizontal axis coordinates from 1 to 160 correspond to centric-out phase-encode lines acquisition order from the central (low frequency \(k_{\text{phase}} = 0\)) to the peripheral region of k-space (high frequency); vertical axis coordinates correspond to the motions of each \(k_{\text{phase}}\) line in one of the three motion directions (two translations with unit pixel and rotation with unit degree). (a) piecewise linear. (b) parabolic. (c) sinusoidal.

The reference image was chosen to be a motion-free complex real-experimental head MR image in the sagittal plane formed by raw k-space data in one coil. The reference image was acquired on a Siemens Verio 3T Scanner by a single-slice 2D GRE sequence with centric-out phase-encode line order at Auburn University MRI Research Center. Since this acquisition order of \(k_{\text{phase}}\) lines was used for real data in the proposed method, the same acquisition order with the reference sequence was used to simulate motion in this section. This made the simulations closer to the real experiments. In each simulation, the input was a motion-corrupted image generated by modeling motion using one of the above motion trajectories; the output was five motion-corrected images with five motion estimation plots.
using the five criteria. In most cases, the proposed method can effectively reduce motion artifacts no matter which criterion was used. In some cases, the difference between the motion-corrected images and the reference image by visual inspection is minimal. Fig 4.11 shows one example of comparison between the reference image, motion-corrupted image, and five motion-corrected images.
Figure 4.11: Comparison of the reference image, motion-corrupted image, and five corrected images. (a) The reference image. (b) The motion-corrupted image. (c) The motion-corrected image based on image entropy. (d) The motion-corrected image based on the entropy of one-directional gradient of image along the phase-encoding direction. (e) The motion-corrected image based on entropy of two directional gradients of image. (f) The motion-corrected image based on total variation. (g) The motion-corrected image based on the sum of absolute value.

In some cases, some criteria performed worse than other criteria. In Fig 4.12, the motion-corrected image based on the sum of absolute value looks worse than other corrected images, and some artifacts cannot be properly erased in that image.
Figure 4.12: Comparison of the reference image, motion-corrupted image, and five motion-corrected images and corresponding motion estimation plots. (a) The reference image. (b) The simulated motion-corrupted image. (c) The motion-corrected image based on the image entropy. (e) The motion-corrected image based on the entropy of the one-directional gradient of image along the phase-encoding direction. (g) The motion-corrected image based on the entropy of the two directional gradients of image. (i) The motion-corrected image based on the total variation. (k) The motion-corrected image based on the sum of the absolute values. (d) (f) (h) (j) (l) are the comparisons of the estimated motion trajectories and real motion trajectories based on the five criteria. The solid curves represent real motion trajectories (green for horizontal motion, red for vertical motion and black for rotation), and the dashed curves represent estimated motion trajectories (green for horizontal translation and red for vertical translation with unit pixel of the y axis; black for rotation with unit degree of the y axis).
Comparing Fig 4.12 (d)(f)(h)(j)(l), Fig 4.12 (l) had the biggest error between real and estimated motion trajectories. That’s the reason why Fig 4.12 (k) still had artifacts.

When the simulated displacement is large, such as a sinusoidal motion trajectory, the motion-corrected images based on the image entropy and the sum of the absolute values were worse than the motion-corrected images based on the entropy of the one-directional gradient of image along the phase-encoding direction, the entropy of the two directional gradients of image and the total variation. Fig 4.13 shows the results.
Figure 4.13: Comparison of the reference image, motion-corrupted image, and five motion-corrected images and corresponding motion estimation plots. (a) The reference image. (b) The simulated motion-corrupted image. (c) The motion-corrected image based on the image entropy. (e) The motion-corrected image based on the entropy of the one-directional gradient of image along the phase-encoding direction. (g) The motion-corrected image based on the entropy of the two directional gradients of image. (i) The motion-corrected image based on the total variation. (k) The motion-corrected image based on the sum of the absolute values. (d) (f) (h) (j) (l) are the comparisons of the estimated motion trajectories and real motion trajectories based on the five criteria. The solid curves represent real motion trajectories (green for horizontal motion, red for vertical motion and black for rotation), and the dashed curves represent estimated motion trajectories (green for horizontal translation and red for vertical translation with unit pixel of the $y$ axis; black for rotation with unit degree of the $y$ axis).

In Fig 4.13 (f) (h) (j), although the estimated motion cannot totally correct the real motion at the high frequency part (the last one quarter of the sinusoidal curves), there is not too much impact on the motion-corrected images. Although the motion-corrected images based on the image entropy and the sum of the absolute values still had some artifacts, the motion-corrected images were improved significantly compared to the simulated motion-corrupted image.
4.9 Experiments

The performance of the three criteria—the entropy of the one or two directional gradients of image and the total variation—under the proposed method were good and similar in the simulations. Therefore, we selected one of the three—the total variation—to be the focus criterion to correct real motion in the experiments. Head images in the sagittal and axial plane were acquired on a Siemens Verio 3T Scanner using the same sequence as in the Simulations section (Auburn University MRI Research Center). The acquisition order of $k_{\text{phase}}$ lines is used to correct the motion. The subject was instructed to perform head nods to acquire motion-corrupted images in the sagittal plane and side-to-side motion to acquire motion-corrupted images in the axial plane. Two no-motion images were acquired separately by the same scan sequence as reference images for comparison. Fig 4.14 and Fig 4.15 show the results of applying the proposed motion correction method on the motion-corrupted images. The motion-corrected images indicate that our method was able to eliminate motion artifacts and preserve sharp boundaries. This method is also efficient. The whole processing time for $160 \times 160$ images is under a minute on an 8GB RAM Intel Core i7-3630QM 2.40 GHz CPU processor machines using MATLAB.
Figure 4.14: Comparison of the reference image, real motion-corrupted image and motion-corrected image in the sagittal plane. Due to the motion difference between the neck part and the head part, the image above the green dashed line part was used to calculate the focus criterion. (a) No-motion (reference) image. (b) Real motion-corrupted image. (c) Motion-corrected image.

Figure 4.15: Comparison of the no-motion image, real motion-corrupted image and motion-corrected image in the axial plane. (a) No-motion (reference) image. (b) Real motion-corrupted image. (c) Motion-corrected image.

For the sagittal plane images, due to the different motion between neck part (below the green dashed line) and head part (above the green dashed line), a selected window (above the green dashed line) designated a FOV over which to calculate the focus criterion, which explains why the neck part of the motion-corrected image is still blurred.
4.10 Discussion

The proposed method on a windowed image may potentially correct some complicated nonrigid motions. In some cases, we can assume that the whole image can be divided into several small subimages, within which the motion can be modeled as rigid. The subimages are combined after motion of each subimage is corrected by the windowed method.

The proposed method can be extended to 3D. For 3D rigid motion, six motion trajectories need to be estimated. Due to the increased search space in 3D, the piecewise linear initial approximation needs to be more accurate and efficient.
Chapter 5
An Efficient Motion Correction Method Based on Fast Robust Correlation

Translational search can be computationally demanding. A correlation operation can be used to calculate an image match when the matching criterion is the sum of squared errors. However, this approach cannot be used for nonquadratic matching criteria. Fast robust correlation is a computationally efficient search algorithm for translational image matching in the frequency domain. This method can calculate matching surfaces from nonquadratic criteria using a series of high-speed correlations by defining a kernel with sinusoidal terms. In this chapter, an efficient motion correction method based on fast robust correlation is proposed. The proposed method corrects motion-distorted images by aligning translational motion between images formed by neighboring frequency segments. Since the squared difference kernel is invariant to motion between partial-Fourier images, we adopt the absolute value kernel, which can be easily approximated by sinusoidal terms. Total variation of the sum of partial-Fourier images is chosen as the new matching criterion. FFTs are used to calculate correlations for computational speed. Different search strategies to combine and correct motion over the whole k-space are discussed and compared. The proposed method can perform real-time processing to reduce image motion artifacts significantly in the simulations and MRI cardiac experiments.

This chapter is organized as follows: 5.1 review of fast robust correlation, 5.2 motion model and acquisition assumption, 5.3 the failure of squared kernel, 5.4 proposed match criterion, 5.5 proposed method to estimate the relative motion between the neighboring segments, 5.6 search strategy, 5.7 simulations, 5.8 experiments, and 5.9 discussion.
5.1 Fast robust correlation review

Fast robust correlation (FRC) was first proposed by Fitch et al. [30]. This method implements a series of correlations to calculate the matching surface to increase computational speed. Robust behavior is achieved by defining a new kernel, which is composed of several sinusoidal terms. FRC is used in image registration, such as template matching [51] and image fusion [106]. Based on the sinusoidal terms kernel [30, 48], some principal component analysis methods [92, 58] have been proposed.

This new motion correction method is inspired by FRC. The main idea of FRC is described first.

A matching surface is used to calculate the match quality of two images over a range of 2-D shifts. The general matching surface is expressed as:

\[ S(m_x, m_y) = \sum_{x,y} h(f(x, y) - g(x - m_x, y - m_y)) \alpha_f(x, y) \alpha_g(x - m_x, y - m_y), \]  

where \( f \) and \( g \) are two images, \( m_x \) and \( m_y \) are two-directional translations, \((x, y)\) is the pixel index, \( \alpha_f(x, y) \) and \( \alpha_g(x, y) \) are image masks, and \( h(r) \) is the difference kernel, where \( r \) is the pixel difference.

Correlation functions can be calculated efficiently with FFTs, and sinusoidal terms can be rewritten as correlations to make FFT implementation possible. A nonquadratic kernel is chosen for its sensitivity to outliers and efficiency.

\[ h(r) = \sum_{p=1}^{P} b_p (1 - \cos(a_p \pi r)), \]
where the parameter sets \(a_p\) and \(b_p\) determine the curve of the kernel. The robust matching surface with this kernel becomes:

\[
S(m_x, m_y) = \sum_{x,y} \alpha_f(x,y)\alpha_g(x-m_x, y-m_y) \sum_{p=1}^{P} b_p \left[ 1 - \cos \left( a_1 \pi \left( f(x,y) - g(x-m_x, y-m_y) \right) \right) \right].
\]  

(5.3)

Eq (5.3) can be expressed as:

\[
S(m_x, m_y) = \Re \left\{ \left( \alpha_f \otimes \alpha_g \right) \sum_{p=1}^{P} b_p - \sum_{p=1}^{P} b_p \left[ \left( \alpha_f \exp(ja_p \pi f) \right) \otimes \left( \alpha_g \exp(ja_p \pi g) \right) \right] \right\},
\]  

(5.4)

where \(\otimes\) represents correlation. Correlations can be efficiently calculated in the Fourier domain by FFTs.

### 5.2 Motion model and assumption

In this chapter, motion is assumed to be two-directional translations in a 2D plane. However, compared with the previous chapter, the basic unit for motion correction is a certain number of \(k_y\) lines, which we refer to as a segment of k-space data. The whole k-space acquisition is assumed to have occurred over time in several non-overlapping neighboring segments, and we only consider the motion between different segments. Within each segment, the object is modeled as stationary. The motion assumption is shown in Figure 5.1. This assumption may not be accurate enough in some cases, but for many problems it is a good approximation.

The motion parameters \(\Delta x\) and \(\Delta y\) are functions of \(k_s\), which is segment \(s\) of \(k_y\) lines. Therefore, the motion correction is essential to estimate a total of \(N_s\) 2D vectors \(M\), where \(N_s\) is the number of segments. The motion parameters are given by:

\[
M(k_s) = (\Delta x(k_s), \Delta y(k_s)),
\]  

(5.5)

where \(\Delta x, \Delta y\) are two-directional translation estimations.
Figure 5.1: The motion assumption for the proposed motion correction method. We only consider the motion between different segments. Within each segment, the object is assumed motionless.

5.3 Squared kernel

The squared difference kernel \( h(r) = r^2 \), where \( r \) is the pixel difference, is not suitable for motion image reconstruction in the context of non-overlapping frequency-domain segments. To see this, consider the matching surface based on the squared difference kernel defined in [30] as follows:

\[
S(m_x, m_y) = \sum_{x,y} \left( f(x, y) - g(x - m_x, y - m_y) \right)^2 \alpha_f(x, y)\alpha_g(x - m_x, y - m_y). \quad (5.6)
\]

Eq (5.6) can be expressed as:

\[
S = (f^2\alpha_f) \otimes \alpha_g - 2(f\alpha_f) \otimes (g\alpha_g) + \alpha_f \otimes (g^2\alpha_g), \quad (5.7)
\]

where \( \otimes \) represents correlation. The first term and the third term of Eq (5.7) are constant and can be neglected, and the second term can be computed in the Fourier domain:

\[
f \otimes g = \text{ifft} \left( \text{fft}(f)\text{fft}^*(g) \right), \quad (5.8)
\]

where \( * \) represents complex conjugate.
Since frequency segments do not overlap, this term is zero for any shifts, and the matching surface Eq (5.6) is constant.

### 5.4 Proposed match criterion

A new criterion is defined in the proposed method. The criterion is the total variation of the sum of images formed by the neighboring k-space segments. Signals with sharp edges have larger total variation in the presence of phase distortion, which is the kind of error that occurs when different frequency segments are acquired at different object positions.

A one-dimensional example illustrates this criterion. The translational motion in the spatial domain is equal to a linear-phase shift in the frequency domain, according to the Fourier shift property. The process of phase shift can be considered as the signal go through the unity gain nonlinear-phase filter. This nonlinear phase can introduce ringing. Figure 5.2 shows a square wave filtered by a unity gain nonlinear-phase filter. The green unfiltered signal represents an image without translational motion and the red filtered signal represents image with translational motion. The green signal has some wave-shape signal disturbance at the sharp edge after filtering. Total variation effectively measures the level of ringing, which can reflect the level of the translation motion. This example illustrates how total variation can be used to detect translation motion even when the frequency segment do not overlap.

As part of the proposed criterion, the absolute value kernel is used:

\[ h(r) = |r| \]  

(5.9)

Correlation functions can be calculated efficiently with FFTs, but correlation assumes a squared kernel.

According to the FRC method, sinusoidal terms are used to approximate the proposed kernel. These terms can then be rewritten as correlations to make FFT implementation
Possible. A Fourier series is used as a new method to calculate the sinusoidal approximation. Since most kernels satisfy the Dirichlet conditions, the Fourier series is guaranteed to converge. For some kernels, these calculations can be relatively easy compared to [30]. Here a Fourier series is used to estimate the absolute value kernel. According to [30], pixel values should be normalized to the range of $[-1, 1]$, so we consider the Fourier series of $h(r)$ on the range of $[-1, 1]$:

$$h(r) = |r| = a_0 + \sum_{p=1}^{\infty} \left( a_p \cos(pr) + b_p \sin(pr) \right),$$  \hspace{1cm} (5.10)

where:

$$\begin{cases} 
  a_0 = \frac{1}{2} \int_{-1}^{1} h(r) dr, \\
  a_p = \int_{-1}^{1} h(r) \cos(pr) dr, p \geq 1, \\
  b_p = \int_{-1}^{1} h(r) \sin(pr) dr, p \geq 1.
\end{cases}$$  \hspace{1cm} (5.11)

Eq (5.9) is even, so $b_p = 0$. The approximation becomes:

$$h(r) \approx \frac{1}{2} - \frac{4}{\pi^2} \sum_{p=1}^{P} \frac{1}{(2p-1)^2} \cos((2p-1)\pi r) = \frac{1}{2} - \sum_{p=1}^{P} B_p \cos(A_p \pi r).$$  \hspace{1cm} (5.12)

This result is similar to [27]. Figure 5.3 compares finite sum approximation with the absolute value kernel. The sinusoidal kernels approach the absolute value kernel as $P$ increases. Based
on the empirical performance, we choose $P = 3$ to approximate the absolute value kernel, which is enough to perform well in most cases.

Figure 5.3: Comparison of different values of $P$ and the absolute value kernel. (a) $P = 1$. (b) $P = 2$. (c) $P = 3$. (d) $P = 4$. (e) $P = 5$. (d) $P = 6$. 
5.5 Proposed method

\( f \) and \( g \) represent two images formed by two neighboring k-space segments. \( f_1 \) and \( f_2 \) are the first difference of \( f \) in the horizontal and vertical directions. The same definition is used for \( g_1 \) and \( g_2 \). Since \( f_1, f_2, g_1, g_2 \) are all complex images, the real and imaginary parts of all these images can be represented as \( f_{1r}, f_{1i}, f_{2r}, f_{2i}, g_{1r}, g_{1i}, g_{2r}, g_{2i} \). The matching surface is the sum of four matching surfaces, which are formed by \( f_{1r} \) and \( g_{1r} \), \( f_{1i} \) and \( g_{1i} \), \( f_{2r} \) and \( g_{2r} \), and \( f_{2i} \) and \( g_{2i} \). Consider \( f_{1r} \) and \( g_{1r} \) as an example to show the whole calculation process. Using Eq (5.12) to get the matching surface:

\[
S_{1r}(m_x, m_y) = \sum_{x,y} \alpha_{f_{1r}}(x,y)\alpha_{g_{1r}}(x-m_x, y-m_y) \times \left[ \frac{1}{2} - \sum_{p=1}^{P} B_p \cos \left( A_p \pi (f_{1r}(x,y) + g_{1r}(x-m_x, y-m_y)) \right) \right].
\]  

(5.13)

Eq (5.13) is equal to:

\[
S_{1r}(m_x, m_y) = \Re \left\{ \frac{1}{2} (\alpha_{f_{1r}} \otimes \alpha_{g_{1r}}) - \sum_{p=1}^{P} B_p (\alpha_{f_{1r}} e^{jA_p \pi f_{1r}} \otimes \alpha_{g_{1r}} e^{-jA_p \pi g_{1r}}) \right\}.
\]

(5.14)

Using FFTs to Eq (5.14):

\[
S_{1r}(m_x, m_y) = \Re \left\{ \text{ifft} \left( \frac{1}{2} \text{fft}(\alpha_{f_{1r}}) \text{fft}^*(\alpha_{g_{1r}}) - \sum_{p=1}^{P} B_p (\text{fft}(\alpha_{f_{1r}} e^{jA_p \pi f_{1r}}) \text{fft}^*(\alpha_{g_{1r}} e^{-jA_p \pi g_{1r}})) \right) \right\}.
\]

(5.15)

After the four matching surfaces are computed, the translations can be calculated as follows:

\[
(\hat{m}_x, \hat{m}_y) = \arg \min_{(m_x, m_y)} (S_{1r}(m_x, m_y) + S_{1i}(m_x, m_y) + S_{2r}(m_x, m_y) + S_{2i}(m_x, m_y)).
\]

(5.16)

To obtain subpixel accuracy, interpolation is implemented by padding zeros in the frequency domain of the partial matching surface.

71
5.6 Search strategy

The sections above illustrated how to calculate the relative motion between two neighboring non-overlapping segments, but we still need search strategies to combine and correct motion over the whole k-space. For each search strategy, there are two kinds of segments — the reference segment and the floating segment. The relative motion between the reference and the floating segment is detected and corrected for each floating segment using the above method until all the segments are corrected for the whole k-space. Based on the empirical performance, we have concluded that no gap in k-space should be allowed between the reference and floating segments to obtain reliable motion estimates. Some search strategies are proposed in this section and the results are compared in the Simulations section. It is assumed that k-space consists of \( n \) segments.

*Strategy 1*

From the top to the bottom of k-space, each segment is chosen to be the reference segment, and the neighboring segment below it is the floating segment. After calculation of the relative motion of \( n - 1 \) pairs of reference and floating segments, the motion relative to a certain segment of all the other segments is now known and is then corrected. The whole process is shown in Figure 5.4.

*Strategy 2*

To take advantage of the previous calculated motion information, the reference segment is enlarged after combining the previous corrected floating segments. The whole process is shown in Figure 5.5.

*Strategy 3*

The segment with zero frequency is chosen as reference, which is enlarged after combining the previous corrected floating segments similar to Strategy 2. Considering the impact of the symmetric segments in k-space, the neighboring symmetric segments are in turn chosen as the floating segments. The whole process is shown in Figure 5.6.
Compared with the Strategy 3, the odd segment is corrected twice. When the odd segment is corrected the first time, the symmetric segments are not included in the reference segment. This omission may result in error. So in Strategy 4, when the odd segments are corrected the second time, the corrected symmetric segments are combined into the reference segment. The whole process is shown in Figure 5.7.

**Strategy 5**

Compared with Strategy 4, the odd and even segments both are corrected twice, which is shown in Figure 5.8 in detail.

---

Figure 5.4: Search Strategy 1 of the proposed method. Different numbers represent different segments. The yellow segment is the reference segment, the light blue one is the corresponding floating segment, and the blue one is the motion-corrected segment. From top to bottom, the relative motion between the reference and floating segments is determined.
Figure 5.5: Search Strategy 2 of the proposed method. Different numbers represent different segments. The blue segment is the reference segment, and the light blue one is the corresponding floating segment. The reference segment is enlarged after combining the previous corrected floating segment until all the segments are corrected.
Figure 5.6: Search Strategy 3 of the proposed method. The blue segment is the reference segment, and the light blue one is the corresponding floating segment. The reference segment is enlarged after combining the previous corrected floating segment until all the segments are corrected. The neighboring symmetric segments are in turn chosen as the floating segments.
Figure 5.7: Search Strategy 4 of the proposed method. The blue segment is the reference segment, the light blue one is the corresponding floating segment and the yellow one is the segment corrected twice. Take segment 3 for example. Segment 3 and 4 are symmetric segments. After segment 3 and segment 4 are corrected in turn, the segment 3 is corrected again with the corrected segment 4 as part of the reference.
Figure 5.8: Search Strategy 5 of the proposed method. The blue segment is the reference segment, the light blue one is the corresponding floating segment and the yellow one is the segment corrected twice. The symmetric segments are corrected twice.
5.7 Simulations

To validate the proposed algorithm and compare the search strategies, simulations were conducted in this section. One motion-free real-experimental MRI gating cardiac image (complex pixels) was chosen as the reference image, whose size is 128 by 120. The reference image was acquired on a clinical 3T Siemens Verio scanner with balanced Steady State Free Precession (bSSFP) sequence 32-channel cardiac array coil at Auburn University MRI Research Center. The frequency domain of the reference image was separately divided evenly into 15, 12, 10 and 8 parts, corresponding to 8, 10, 12 and 15 lines in each segment, to simulate segments in k-space. Different search strategies of the proposed motion correction method were implemented to detect and correct the simulated translations. Two random values in the range from -10 to 10 pixels, which are big enough to model real motion, were generated to simulate two translations in each segment except the reference segment.

In this section, different search strategies are compared and some motion correction examples are shown.

5.7.1 comparison of search strategies

For each segment size (total of 4 different sizes), 10,000 random translational motion images were generated by the above processing. The 10,000 simulated images were separately processed by the above five search strategies, and the results were compared by the average estimation error for each segment and the average values of four image quality criteria. The motion correction accuracy was $\frac{1}{5}$ pixel here.

The average estimation error for segment $j$ is given by:

$$\Delta x_j = \frac{1}{10000} \sum_{i=1}^{10000} |x_{i,j} - \hat{x}_{i,j}|, \Delta y_j = \frac{1}{10000} \sum_{i=1}^{10000} |y_{i,j} - \hat{y}_{i,j}|,$$  \hspace{1cm} (5.17)
where $\Delta x_j$ and $\Delta y_j$ are the average estimation error of segment $j$ in two-translational directions, $x_{i,j}$ and $y_{i,j}$ are the actual translations of the segment $j$ from image $i$ in two-translational directions, and $\hat{x}_{i,j}$ and $\hat{y}_{i,j}$ are the estimated translations of the segment $j$ from image $i$ in two-translational directions using the proposed method.

The average values of four image quality criteria were calculated—SSD (Sum of Squared Difference), IE (image entropy), EG (entropy of gradients of an image) and TV (total variation of an image). SSD (Eq (5.18)) compares one image with the reference image. The definition of image entropy is given by Eq (4.2) and Eq (4.3). The definition of the entropy of gradients of an image is given by Eq (4.5) and Eq (4.6). The definition of the total variation of an image is given by Eq (4.7) and Eq (4.8).

SSD is given by:

$$SSD = \sum_{x=1}^{N} |I(x) - I_{\text{ori}}(x)|^2,$$

where $x$ is the pixel index and $N$ is the total number of pixels of the image.

For each segment size, the average estimation error for each segment and the average values of four image quality criteria, are given to compare the results. Table 5.1 to Table 5.8 show the results of different segment sizes.

Note: $I_{\text{ref}}$ represents the reference image, $I_{\text{mot}}$ represents the motion-corrupted image, $I_{\text{rec1}}$ represents the reconstructed image using Strategy 1, $I_{\text{rec2}}$ represents the reconstructed image using Strategy 2, $I_{\text{rec3}}$ represents the reconstructed image using Strategy 3, $I_{\text{rec4}}$ represents the reconstructed image using Strategy 4 and $I_{\text{rec5}}$ represents the reconstructed image using Strategy 5. In the tables of average estimation error, the best entry of each row is marked as boldface. In the tables of average values of four criterion, the best entry of each column is marked as boldface. The motion-corrupted image comes from placing original random translational segments together, which is given by:

$$I_{\text{mot}} = \text{iffh} \left( \sum_{i=1}^{N} F_i(k, l) \right) \quad (5.19)$$
where $F_i(k, l)$ is the $i$th segment and $N$ is the total number of segments.

Table 5.1: Average translational motion estimation errors (two directions $\Delta_x, \Delta_y$) of each k-space segment (8 lines) for each search strategy (pixel/unit)

<table>
<thead>
<tr>
<th>Segment</th>
<th>Strategy 1</th>
<th>Strategy 2</th>
<th>Strategy 3</th>
<th>Strategy 4</th>
<th>Strategy 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0998, 1.8990</td>
<td>0.0998, 1.8990</td>
<td>0.0650, 1.1691</td>
<td>0.0615, 0.6923</td>
<td>0.0644, 0.5356</td>
</tr>
<tr>
<td>2</td>
<td>4.9890, 5.0110</td>
<td>0.1774, 1.8724</td>
<td>0.0603, 0.4606</td>
<td>0.0602, 0.4606</td>
<td>0.0558, 0.3620</td>
</tr>
<tr>
<td>3</td>
<td>5.0539, 5.0143</td>
<td>0.1291, 2.0006</td>
<td>0.0736, 1.2740</td>
<td>0.1468, 0.4411</td>
<td>0.1537, 0.3837</td>
</tr>
<tr>
<td>4</td>
<td>5.0318, 5.0687</td>
<td>0.1597, 2.1334</td>
<td>0.0692, 0.4621</td>
<td>0.0637, 0.3783</td>
<td>0.0649, 0.2570</td>
</tr>
<tr>
<td>5</td>
<td>5.0217, 4.9801</td>
<td>0.1831, 2.5413</td>
<td>0.0789, 0.9800</td>
<td>0.1153, 0.2986</td>
<td>0.1202, 0.3256</td>
</tr>
<tr>
<td>6</td>
<td>5.0286, 4.9957</td>
<td>0.1785, 3.1942</td>
<td>0.0814, 0.4605</td>
<td>0.0980, 0.2475</td>
<td>0.0998, 0.1887</td>
</tr>
<tr>
<td>7</td>
<td>4.9305, 5.0669</td>
<td>0.1680, 3.8557</td>
<td>0.0961, 0.7970</td>
<td>0.0713, 0.2394</td>
<td>0.0766, 0.3029</td>
</tr>
<tr>
<td>8</td>
<td>5.0587, 5.0004</td>
<td>0.1600, 3.4528</td>
<td>0.1398, 0.4593</td>
<td>0.1731, 0.2128</td>
<td>0.1647, 0.1852</td>
</tr>
<tr>
<td>9</td>
<td>4.9750, 5.0455</td>
<td>0.1703, 3.0327</td>
<td>0.0985, 0.7714</td>
<td>0.1754, 0.2096</td>
<td>0.1844, 0.3078</td>
</tr>
<tr>
<td>10</td>
<td>4.9994, 5.0006</td>
<td>0.1929, 2.7654</td>
<td>0.1880, 0.4545</td>
<td>0.2362, 0.1895</td>
<td>0.2154, 0.2011</td>
</tr>
<tr>
<td>11</td>
<td>4.9730, 4.9784</td>
<td>0.2079, 2.5063</td>
<td>0.0894, 0.9387</td>
<td>0.1939, 1.2875</td>
<td>0.2008, 1.0000</td>
</tr>
<tr>
<td>12</td>
<td>5.0472, 4.9806</td>
<td>0.2630, 1.5380</td>
<td>0.2202, 0.5570</td>
<td>0.2076, 0.7596</td>
<td>0.2196, 0.5980</td>
</tr>
<tr>
<td>13</td>
<td>5.0237, 4.9882</td>
<td>0.2729, 1.6419</td>
<td>0.2018, 0.9620</td>
<td>0.1934, 0.8191</td>
<td>0.2392, 1.1085</td>
</tr>
<tr>
<td>14</td>
<td>4.9548, 5.0724</td>
<td>0.2869, 1.4097</td>
<td>0.2664, 0.6443</td>
<td>0.2977, 1.8543</td>
<td>0.3055, 1.4656</td>
</tr>
</tbody>
</table>

Table 5.2: Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (8 lines segment)

<table>
<thead>
<tr>
<th>Images</th>
<th>SSD</th>
<th>IE</th>
<th>EG</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{ref}$</td>
<td>0</td>
<td>410.1539</td>
<td>879.8204</td>
<td>1.1352e-3</td>
</tr>
<tr>
<td>$I_{mot}$</td>
<td>1.8313e-10</td>
<td>448.4703</td>
<td>979.2290</td>
<td>1.2628e-3</td>
</tr>
<tr>
<td>$I_{rec1}$</td>
<td>1.3378e-10</td>
<td>415.4935</td>
<td>899.3643</td>
<td>1.1604e-3</td>
</tr>
<tr>
<td>$I_{rec2}$</td>
<td>1.1768e-10</td>
<td>413.3635</td>
<td>891.7397</td>
<td>1.1487e-3</td>
</tr>
<tr>
<td>$I_{rec3}$</td>
<td>1.3218e-11</td>
<td>411.9923</td>
<td>882.5268</td>
<td>1.1375e-3</td>
</tr>
<tr>
<td>$I_{rec4}$</td>
<td>5.2190e-12</td>
<td>410.7653</td>
<td>881.7578</td>
<td>1.1367e-3</td>
</tr>
<tr>
<td>$I_{rec5}$</td>
<td>4.6205e-12</td>
<td>410.501</td>
<td>881.8877</td>
<td>1.1367e-3</td>
</tr>
</tbody>
</table>

From Table 5.1 to Table 5.8, we can draw the following observations:

1. Comparing Strategy 1 with Strategy 2, the results of Strategy 2 have smaller average translational motion estimation errors and the average values of criteria. Average translational motion estimation errors for Strategy 1 are extraordinarily large, because the size of the reference segments does not include enough data to get accurate estimation and the reference segments are not extended.
Table 5.3: Average translational motion estimation errors (two directions $\Delta x, \Delta y$) of each k-space segment (10 lines) for each search strategy (pixel/unit)

<table>
<thead>
<tr>
<th>Segment</th>
<th>Strategy 1</th>
<th>Strategy 2</th>
<th>Strategy 3</th>
<th>Strategy 4</th>
<th>Strategy 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2134,0.2966</td>
<td>0.2134,0.2966</td>
<td><strong>0.0674,0.3207</strong></td>
<td>0.0674,0.3207</td>
<td>0.0674,0.3207</td>
</tr>
<tr>
<td>2</td>
<td>4.9520,4.9737</td>
<td>0.2863,0.3403</td>
<td>0.0609,0.3720</td>
<td><strong>0.0602,0.1781</strong></td>
<td>0.0602,0.1781</td>
</tr>
<tr>
<td>3</td>
<td>5.0099,5.0004</td>
<td>0.2869,0.3924</td>
<td>0.0748,0.1590</td>
<td>0.0748,0.1590</td>
<td><strong>0.0694,0.1619</strong></td>
</tr>
<tr>
<td>4</td>
<td>4.9899,5.0173</td>
<td>0.2787,0.5175</td>
<td>0.0709,0.4382</td>
<td><strong>0.0735,0.2651</strong></td>
<td>0.0760,0.2639</td>
</tr>
<tr>
<td>5</td>
<td>4.9925,4.9691</td>
<td>0.2745,0.6683</td>
<td>0.0963,0.1843</td>
<td>0.0943,0.1652</td>
<td><strong>0.0808,0.1736</strong></td>
</tr>
<tr>
<td>6</td>
<td>4.9876,5.0228</td>
<td>0.2535,1.3170</td>
<td>0.1051,0.4581</td>
<td><strong>0.1111,0.3079</strong></td>
<td>0.1185,0.3084</td>
</tr>
<tr>
<td>7</td>
<td>5.0102,5.0101</td>
<td>0.2353,0.9771</td>
<td>0.1338,0.3003</td>
<td>0.1255,0.2373</td>
<td><strong>0.1089,0.2459</strong></td>
</tr>
<tr>
<td>8</td>
<td>5.0057,5.0135</td>
<td>0.2172,0.7450</td>
<td>0.1279,0.4857</td>
<td>0.1191,0.3547</td>
<td><strong>0.1261,0.3562</strong></td>
</tr>
<tr>
<td>9</td>
<td>4.9760,5.0224</td>
<td>0.2170,0.6700</td>
<td>0.0897,0.3600</td>
<td><strong>0.0901,0.3016</strong></td>
<td>0.0903,0.3049</td>
</tr>
<tr>
<td>10</td>
<td>5.0127,5.0346</td>
<td>0.2144,0.6059</td>
<td>0.2131,0.4807</td>
<td><strong>0.1898,0.4595</strong></td>
<td>0.2106,0.4675</td>
</tr>
<tr>
<td>11</td>
<td>5.0420,5.0565</td>
<td>0.1734,0.6280</td>
<td><strong>0.1343,0.4503</strong></td>
<td>0.1249,0.6279</td>
<td>0.1243,0.6196</td>
</tr>
</tbody>
</table>

Table 5.4: Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (10 lines segment)

<table>
<thead>
<tr>
<th>Images</th>
<th>SSD</th>
<th>IE</th>
<th>EG</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{\text{ref}}$</td>
<td>0</td>
<td>410.1539</td>
<td>879.8204</td>
<td>1.1352e-3</td>
</tr>
<tr>
<td>$I_{\text{mot}}$</td>
<td>5.4738e-11</td>
<td>434.0825</td>
<td>972.0623</td>
<td>1.2550e-3</td>
</tr>
<tr>
<td>$I_{\text{rec1}}$</td>
<td>2.1977e-11</td>
<td>413.5135</td>
<td>892.3260</td>
<td>1.1504e-3</td>
</tr>
<tr>
<td>$I_{\text{rec2}}$</td>
<td>5.8409e-12</td>
<td>409.5080</td>
<td>882.7683</td>
<td>1.1365e-3</td>
</tr>
<tr>
<td>$I_{\text{rec3}}$</td>
<td>4.7798e-12</td>
<td>410.7949</td>
<td>880.7689</td>
<td>1.1350e-3</td>
</tr>
<tr>
<td>$I_{\text{rec4}}$</td>
<td><strong>2.7349e-12</strong></td>
<td><strong>410.2985</strong></td>
<td><strong>880.5240</strong></td>
<td><strong>1.1348e-3</strong></td>
</tr>
<tr>
<td>$I_{\text{rec5}}$</td>
<td>2.7509e-12</td>
<td>410.3011</td>
<td>880.6121</td>
<td>1.1349e-3</td>
</tr>
</tbody>
</table>

Table 5.5: Average translational motion estimation errors (two directions $\Delta x, \Delta y$) of each k-space segment (12 lines) for each search strategy (pixel/unit)

<table>
<thead>
<tr>
<th>Segment</th>
<th>Strategy 1</th>
<th>Strategy 2</th>
<th>Strategy 3</th>
<th>Strategy 4</th>
<th>Strategy 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1197,0.1178</td>
<td><strong>0.1197,0.1178</strong></td>
<td>0.0758,0.4003</td>
<td>0.0758,0.4003</td>
<td>0.0758,0.4003</td>
</tr>
<tr>
<td>2</td>
<td>5.0288,5.0470</td>
<td>0.1910,0.1925</td>
<td>0.0543,0.3103</td>
<td><strong>0.0545,0.1863</strong></td>
<td><strong>0.0545,0.1863</strong></td>
</tr>
<tr>
<td>3</td>
<td>4.9700,5.0472</td>
<td><strong>0.2726,0.2728</strong></td>
<td>0.0798,0.8030</td>
<td>0.0798,0.8030</td>
<td>0.0840,0.7474</td>
</tr>
<tr>
<td>4</td>
<td>4.9429,4.9447</td>
<td>0.2548,0.3451</td>
<td>0.0812,0.4885</td>
<td>0.0809,0.3721</td>
<td><strong>0.0843,0.3635</strong></td>
</tr>
<tr>
<td>5</td>
<td>5.0067,4.9646</td>
<td>0.2083,1.5289</td>
<td>0.0800,0.8216</td>
<td>0.0790,0.7933</td>
<td><strong>0.0752,0.7521</strong></td>
</tr>
<tr>
<td>6</td>
<td>5.0066,5.0320</td>
<td>0.2108,1.2325</td>
<td>0.1253,0.6003</td>
<td>0.1174,0.4844</td>
<td><strong>0.1215,0.4728</strong></td>
</tr>
<tr>
<td>7</td>
<td>5.0257,5.0153</td>
<td>0.2073,0.9326</td>
<td>0.1058,0.7955</td>
<td>0.1098,0.7434</td>
<td><strong>0.1104,0.6950</strong></td>
</tr>
<tr>
<td>8</td>
<td>4.9647,4.9940</td>
<td>0.2034,0.7582</td>
<td>0.1606,0.5752</td>
<td>0.1546,0.4651</td>
<td><strong>0.1638,0.4525</strong></td>
</tr>
<tr>
<td>9</td>
<td>5.0350,5.0293</td>
<td><strong>0.1342,0.6749</strong></td>
<td>0.1423,0.9133</td>
<td>0.1308,0.8065</td>
<td>0.1295,0.7570</td>
</tr>
</tbody>
</table>

81
Table 5.6: Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (12 lines segment)

<table>
<thead>
<tr>
<th>Images</th>
<th>SSD</th>
<th>IE</th>
<th>EG</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{ref}$</td>
<td>0</td>
<td>410.1539</td>
<td>879.8204</td>
<td>1.1352e-3</td>
</tr>
<tr>
<td>$I_{mot}$</td>
<td>4.6843e-11</td>
<td>430.2335</td>
<td>967.2813</td>
<td>1.2486e-3</td>
</tr>
<tr>
<td>$I_{rec1}$</td>
<td>2.1536e-11</td>
<td>411.6085</td>
<td>889.5606</td>
<td>1.1473e-3</td>
</tr>
<tr>
<td>$I_{rec2}$</td>
<td>7.3393e-12</td>
<td>409.7032</td>
<td>882.7113</td>
<td>1.1357e-3</td>
</tr>
<tr>
<td>$I_{rec3}$</td>
<td>9.0457e-12</td>
<td>409.7552</td>
<td>881.0210</td>
<td>1.1347e-3</td>
</tr>
<tr>
<td>$I_{rec4}$</td>
<td>7.4486e-12</td>
<td>409.8498</td>
<td>880.8988</td>
<td>1.1343e-3</td>
</tr>
<tr>
<td>$I_{rec5}$</td>
<td>6.9332e-12</td>
<td>409.8401</td>
<td>880.6265</td>
<td>1.1341e-3</td>
</tr>
</tbody>
</table>

Table 5.7: Average translational motion estimation errors (two directions $\Delta x, \Delta y$) of each k-space segment (15 lines) for each search strategy (pixel/unit)

<table>
<thead>
<tr>
<th>Segment</th>
<th>Strategy 1</th>
<th>Strategy 2</th>
<th>Strategy 3</th>
<th>Strategy 4</th>
<th>Strategy 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0627,0.0546</td>
<td>0.0627,0.0546</td>
<td>0.0729,1.3594</td>
<td>0.0729,1.3594</td>
<td>0.0729,1.3594</td>
</tr>
<tr>
<td>2</td>
<td>5.0165,5.0098</td>
<td>0.0827,0.1088</td>
<td>0.0552,0.1457</td>
<td>0.0544,0.1419</td>
<td>0.0544,0.1419</td>
</tr>
<tr>
<td>3</td>
<td>4.9549,5.0018</td>
<td>0.1078,0.1671</td>
<td>0.0685,1.0903</td>
<td>0.0685,1.0903</td>
<td>0.0679,1.0800</td>
</tr>
<tr>
<td>4</td>
<td>5.0214,5.1141</td>
<td>0.1545,1.0540</td>
<td>0.0891,0.3546</td>
<td>0.0654,0.3474</td>
<td>0.0654,0.3453</td>
</tr>
<tr>
<td>5</td>
<td>5.0166,4.9946</td>
<td>0.1453,0.8106</td>
<td>0.0843,0.9565</td>
<td>0.0845,0.9556</td>
<td>0.0922,0.9276</td>
</tr>
<tr>
<td>6</td>
<td>4.9673,4.9747</td>
<td>0.1174,0.5954</td>
<td>0.1035,0.3821</td>
<td>0.0859,0.3841</td>
<td>0.0844,0.3785</td>
</tr>
<tr>
<td>7</td>
<td>4.9842,4.9900</td>
<td>0.1451,0.5392</td>
<td>0.1112,0.8694</td>
<td>0.1185,0.8673</td>
<td>0.1284,0.8472</td>
</tr>
</tbody>
</table>

Table 5.8: Average values of four criteria of the reference image, the motion blurred images and the five reconstructed images from the five search strategies (15 lines segment)

<table>
<thead>
<tr>
<th>Images</th>
<th>SSD</th>
<th>IE</th>
<th>EG</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{ref}$</td>
<td>0</td>
<td>410.1539</td>
<td>879.8204</td>
<td>1.1352e-3</td>
</tr>
<tr>
<td>$I_{mot}$</td>
<td>3.9798e-11</td>
<td>428.3945</td>
<td>960.9661</td>
<td>1.2405e-3</td>
</tr>
<tr>
<td>$I_{rec1}$</td>
<td>2.3701e-11</td>
<td>413.3896</td>
<td>895.6266</td>
<td>1.15472e-3</td>
</tr>
<tr>
<td>$I_{rec2}$</td>
<td>1.0784e-11</td>
<td>410.1553</td>
<td>883.0794</td>
<td>1.1370e-3</td>
</tr>
<tr>
<td>$I_{rec3}$</td>
<td>1.0343e-11</td>
<td>410.8307</td>
<td>882.4583</td>
<td>1.1356e-3</td>
</tr>
<tr>
<td>$I_{rec4}$</td>
<td>1.0309e-11</td>
<td>410.8349</td>
<td>882.2986</td>
<td>1.1353e-3</td>
</tr>
<tr>
<td>$I_{rec5}$</td>
<td>1.0138e-11</td>
<td>410.8410</td>
<td>882.1629</td>
<td>1.1351e-3</td>
</tr>
</tbody>
</table>
2. The performance of Strategy 3 is better than Strategy 2, especially when the segment size is smaller. Even if average translational motion estimation errors of Strategy 2 are smaller when the segment size is larger, the difference of the average values of criteria between Strategy 2 and Strategy 3 is trivial. The reference starting from the low frequency contains more energy than the reference starting from the high frequency, which may explain the different performance between Strategy 2 and Strategy 3.

3. Strategy 4 and Strategy 5 always give the best results but with longer processing time, due to the second-time correction of some segments.

4. Strategy 3 gives similar but a little worse performance compared to Strategy 4 and Strategy 5. But Strategy 3 is more efficient. Strategy 3 is used for the next subsection and the Experiments section.

5. Some TV (total variation) values of the corrected images are smaller than the values of the reference image, which illustrates that the total variation criterion is not perfect. To our knowledge, there are no perfect criteria. If the criterion is replaced by other criteria, the results are similar but with higher computational load using more sinusoidal terms.

6. The average translational motion estimation errors may include three parts. The first part error comes from the non-ideal criterion, and the motion may be over-corrected by the criterion. The second part error arises from by the $\frac{1}{3}$ pixel motion correction accuracy of the proposed method. The last part error comes from non-perfect search strategy.

5.7.2 examples of the proposed method

In this subsection, four simulated motion-corrupted images generated by different segment sizes were corrected and reconstructed by the proposed method using search strategy 3. For each example, five sub-figures are included: (a)(b) the real and estimated motions combined with errors in horizontal and vertical directions, (c) the reference image, (d) the motion-corrupted image, (e) the reconstructed image. Figure 5.9 to Figure 5.12 show the four examples.
Figure 5.9: Example 1 of the proposed method with segment size 8.
Figure 5.10: Example 2 of the proposed method with segment size 10.
Figure 5.11: Example 3 of the proposed method with segment size 12.
Figure 5.12: Example 4 of the proposed method with segment size 15.
Subfigures (a)(b) from the figures above illustrate that both estimated translations follow the actual translations, and the error is relative small. The horizontal translations are always better estimated than the vertical motion because the segment contains all the frequency components in the horizontal direction but only partial components in the vertical direction. The errors in the vertical direction mainly just shift the whole image a little.

Subfigures (e) from the figures above show the reconstructed image quality is noticeably improved by the proposed motion correction algorithm and the reconstructed images are almost the same with the reference image. All of these results show successful motion correction for any segment size.

5.8 Experiments

For validation of the proposed algorithm, MRI cardiac imaging experiments were conducted. Cardiac imaging consists of imaging of different cardiac phases. An external electrocardiogram (ECG) signal is a conventional tool to track the phase of the cardiac cycle during imaging. To reduce the effect of heart motion, the acquisition window for each cardiac phase must be short to decrease heart motion during the acquisition, which means that full k-space data cannot be collected in one cardiac cycle. Therefore, multiple cycles are needed to collect the data in the segment pattern, which consists of a certain number of phase lines. For each cardiac phase, the whole k-space data is collected from several cardiac cycles at the same cardiac phase. So only considering heart motion, we can treat cardiac phase imaging motionless due to the short acquisition window and quasi-periodicity of heartbeat. Breathing, which amounts to imposing a motion on top of heart motion, is a big challenge for cardiac imaging. This imposed motion can distort the image. Here we assume that the breathing can be approximated as two translations. Therefore, free-breathing cardiac imaging can be treated as the translational motion correction of each k-space segment, and breath-hold image without other motion can be treated as a no-motion reference image.
In this section, the scan sequence was the same as in the simulation section. The proposed method was used to reconstruct free-breathing real-experimental MRI gating cardiac data (Auburn University MRI Research Center) in each cardiac phase. The original images were distorted due to free-breathing motion. According to the record of the ECG device, the segment size of the frequency domain was chosen as 8 frequency (k-space) lines. Figure 5.13 shows the comparisons of breath-hold no-motion images, original free-breathing images, and reconstructed images from the original free-breathing images for two cardiac phases. The breath-hold no-motion images were acquired separately as reference images for comparison.

Figure 5.13: Comparison of breath-hold no-motion images, original free-breathing images and reconstructed images from the original free-breathing images for two cardiac phases. The first row is one cardiac phase, and the second row is the other cardiac phase. (a)(d) breath-hold no-motion images. (b)(e) original free-breathing images. (c)(f) reconstructed images from the original free-breathing images.
5.9 Discussion

Fig 5.13 shows that the reconstructed images reduce the blurs of the original images effectively, and more details are recovered in the heart region. However, some artifacts remain, especially in the region around the heart. We think the real-world breathing motion gives rise to the remaining artifacts. Real breathing motion is not exactly a two-directional translation in a 2D plane. It includes some through-plane and non-rigid motion, which is beyond the scope of the proposed method.

The proposed method is efficient. The whole process time was around 0.8 s implemented in MATLAB on an 8 GB RAM Intel Core i7-3630QM 2.40 GHz CPU processor machine. Thus, it can be used in real-time processing.
Chapter 6

Motion Correction of Magnitude MR Images using Generative Adversarial Networks

Although the previous proposed methods can correct motion effectively and efficiently, they both require complex-valued raw data. However, raw data is not usually preserved in a clinical environment. In this case the previous two methods cannot be used. In this chapter, we present a novel data-driven motion correction method for magnitude MR images using generative adversarial networks (GANs). GANs (pix2pix model) are implemented to reduce motion artifacts and reconstruct motion-corrupted images through adversarial training between generator and discriminator networks to estimate a motion-corrected image that is close to the reference image. The training set is made of image pairs consisting of motionless reference images and corresponding motion-simulated images. The proposed method is validated by a simulated motion test set and a real motion (experimental) test set.

This chapter is organized as follows: 6.1 introduction to artificial neural networks, 6.2 introduction to convolutional neural networks, 6.3 generative adversarial networks (pix2pix model) basics, 6.4 dataset and training, and 6.5 results.

6.1 Artificial neural networks

Artificial neural networks (ANNs) are widely used in machine learning. In general, an ANN is made of a collection of connected computational units, called artificial neurons, which simulates biological neurons. Artificial neurons are arranged in layers. Data enters a neural network at the input layer, and predictions of the network are generated at the output layer. One or more hidden layers exist between the input and output layer. In general, the basic implementation of a neural network is to provide reasonable predictions by training a
set of labeled data, called training data. During a training process, parameters of a neural network (weights) are tuned by comparing predictions of training data feeding through the network and actual labeled training data, until the network can provide good predictions of the training data. The training process can also be considered as learning a data pattern. After the data pattern is learned, the network is ready to give predictions for unseen data (non-labeled data). Figure 6.1 shows a multilayer perceptron neural network.

Figure 6.1: One multilayer perceptron neural network. The blue neurons represent the neurons from input layer, the green neurons represent the neurons from hidden layers, and the purple neuron represent the neurons from output layer.

For ANNs, at least two types of data sets are needed—training set and test set. A training set is the set of examples used for learning. In this dataset, input data are associated with expected output. A test set is the set of examples used to evaluate the performance of a trained neural network. Sometimes a neural network may require a validation set, but we will only use training set and test set in this chapter.

A basic training process of the multilayer perceptron neural network is discussed here. The process consists of two parts—forward propagation and backward propagation.
For forward propagation, the output vector $\hat{y}$ is calculated layer by layer from the input vector $X$:

$$\hat{y} = (f_n, f_{n-1}, \ldots, f_1)(X; W), \quad (6.1)$$

where $f_n, f_{n-1}, \ldots, f_1$ are activation functions and $W$ is the weight matrix. For each layer, two kinds of calculation are conducted—a linear transformation of the output of the layer and a followed nonlinear function called the activation function. Consider the $k_{th}$ layer as an example.

$$z^{(k)} = o^{(k-1)}W^{(k-1)}, \quad (6.2)$$

where $z^{(k)}$ is the linear transformation of the output of the previous layer, $o^{(k-1)}$ is the output of the $(k-1)_{th}$ layer, and $W^{(k-1)}$ is the weight vector from the $(k-1)_{th}$ layer to the $k_{th}$ layer. The output of the activation function is given by:

$$o^{(k)} = f_k(z^{(k)}), \quad (6.3)$$

where $o^{(k)}$ is the output of the $k_{th}$ layer. Sigmoid, Tanh, ReLU, and Leaky ReLU are the commonly used activation functions.

After $\hat{y}$ is calculated from forward propagation, the loss function is needed to measure the difference between $\hat{y}$ and the labeled output. The task of the training process is to update weights to optimize the prediction of the neural network, which is equal to minimizing the loss function. A backward propagation gradient descent method from the output layer to the input layer is needed. By using the chain rule of derivatives, the gradients of the loss function with respect to all the weights are calculated, and all the weights are updated using gradient descent to minimize the loss function. This is the basic training process for ANNs.

Due to the high computational load, the development of ANNs was slow. But recently, thanks to parallel computation processors (GPUs), ANNs have developed more rapidly.
6.2 Convolutional neural networks

In general, for image neural networks, each pixel of the image is used as the input of the network. If the neurons are fully connected, the computational load is very high, which is not acceptable for typical image sizes. In addition, the huge number of weights will result in overfitting easily. In order to prune the extra connections, convolutional neural networks (CNNs) have been proposed to preserve the spatial relationships of the input images. CNNs are built with a layer structure, and input is fed through layer by layer with neurons in each layer only calculating a small part of the previous layer [62]. CNNs consist of four types of layers: convolutional layer, activation layer, pooling layer, and fully-connected layer. Each type of layer has certain basic functions. A simple CNN structure is shown in Figure 6.2.

![Figure 6.2: Example of a convolutional neural network. (Figure 6.2 is from [62].)](image)

**convolutional layer**

The convolutional layer is used to compute the convolution between a small size filter and a local region of the input layer or the output activation from the previous layer. The
small size filter \((3 \times 3 \text{ or } 5 \times 5)\) slides all over the input layer or the output activation from the previous layer to detect the same feature from the whole image. One convolutional layer may consist of multiple filters with different weights (parameters) in each filter, and each filter performs the above calculation. Each filter presents a feature of the input image, like a horizontal line or corner. Due to the structure where the same filter slides over the whole image, the number of weights needed to be trained is drastically reduced.

*activation layer*

The output from the convolutional layer is fed through the activation layer by calculating element by element. The most popular activation function is ReLU, defined as \(\text{max}(0, x)\).

*pooling layer*

The pooling layer is used to perform a downsampling operation, which is always inserted between the successive convolutional layers. The max pooling, which is the most common pooling operation, preserves a max value from each neighboring \(2 \times 2\) region, and discards 75% of the activations. The pooling operation preserves the important features while reducing the size of the feature map.

*fully connected layer*

All the activations from the previous layer are fully connected by the fully connected layer. The fully connected layer is the same as the ANNs discussed in the previous section. It generates the final output.

### 6.3 Generative adversarial networks (pix2pix model)

Generative adversarial networks (GANs) have been very successful since they were introduced by Goodfellow et al. [35] in 2014. GANs are widely used in generating photorealistic images [111], implementing super-resolution [55], and modeling patterns of motion in video [94] [84]. GANs include two models: a generator model \(G\) and a discriminator model \(D\). The generator model \(G\) tries to generate model distribution \(G(z)\) from random noise \(z\) to match the actual data distribution \(p_{\text{data}}\). The discriminator \(D\) tries to learn to distinguish
the input from the model distribution $G(z)$ or the actual data distribution. The objective of $G$ is to fool $D$, and to make $D$ make a mistake. The training process of GANs is like a fight between the generator and the discriminator. An example of a GAN is shown in Figure 6.3.

![Figure 6.3: An example of a GAN. (Figure 6.2 is from https://skymind.ai/wiki/generative-adversarial-network-gan.)](image)

Our task is to make the generator’s distribution $p_g$ equal to the actual data distribution $p_{\text{data}}$. Input random noise is represented as $z$ with density $p_z(z)$, and the data space generated by the generator is represented as $G(z)$. $D(x)$ is a scalar, which represents the probability that a sample is from the actual data $x$ with density $p_{\text{data}}$ rather than $G(z)$. $D$ is trained to provide the maximum probability to label correctly the actual data $x$ versus the generated data $G(z)$, and $G$ is trained to minimize $\log(1 - D(G(z)))$, so the value function $V(G,D)$ can be represented mathematically as [35]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] . \quad (6.4)$$

The objective function is given by:

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] . \quad (6.5)$$
The global optimum for Eq 6.4 is \( p_g = p_{\text{data}} \), which accomplish our objective. For a detailed discussion of the training process and proof, see [35].

For an unconditional GAN, there are no restrictions or control over the generated data, which may introduce large errors in some cases. Conditional generative adversarial nets (cGANs) were proposed to handle this problem [70] by providing some extra information \( y \) to both generator and discriminator. This additional \( y \) constrains the generator to generate a restricted class of data. An example of a cGAN is given in Figure 6.4.

![cGAN Diagram](image)

Figure 6.4: An example of a cGAN. The discriminator \( D \) is trained to distinguish the tuple between motion-corrected image \( G(y) \) and motion-corrupted image \( y \) from the tuple between motionless image \( x \) and motion-corrupted image \( y \). The generator \( G \) is trained to fool the discriminator \( D \). Both the generator \( G \) and discriminator \( D \) can access the motion corrupted image \( y \) in the cGAN.

The objective function of cGANs is given by [50]:

\[
\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{y,x} \left[ \log D(y, x) \right] + \mathbb{E}_{y,z} \left[ \log(1 - D(y, G(y, z))) \right],
\]  

(6.6)

where \( D \) tries to maximize the objective function and \( G \) tries to minimize it, so the desired \( G^* \) is given by [50]:

\[
G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D).
\]  

(6.7)
The pix2pix model adds another term to the objective function Eq (6.7):

\[ \mathcal{L}_{L^1}(G) = \mathbb{E}_{x,y,z} [||x - G(y,z)||_1], \]  

so the final objective function is given by [50]:

\[
G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L^1}(G).
\]  

The \( L^1 \) term does not change discriminator’s function, but it forces the generated data from the generator to be close to \( x \) (ground truth) in an \( L^1 \) way.

### 6.4 Dataset and training

A pix2pix model requires pairs of images as the training set: motionless images and corresponding motion-corrupted images. 8000 2D images were acquired from 3D datasets of 276 subjects in the Autism Brain Imaging Data Exchange (ABIDE) dataset [23] to be motionless images in the training set. The corresponding motion-corrupted dataset was generated from these motionless images. Motion was simulated in k-space, according to the properties of the Fourier Transform. Each k-space line (along phase encode direction), was modeled with two random translations within \( \pm 12 \) pixels and one random rotation angle within \( \pm 12^\circ \). The center k-space phase lines (randomly chosen from 20 to 60 lines) were preserved without motion corruption to keep the basic image structure.

Two test sets were used to validate the proposed method. One test set—the simulation test set—includes 1780 simulated motion-corrupted images, which were generated by modeling motion of no-motion images in another subset of the ABIDE dataset (53 subjects). The other test set—the experimental test set—includes 200 real motion-corrupted images from two subjects. Six datasets were acquired by a standard vendor-supplied MP-RAGE sequence with resolution 1 mm isotropic and flip angle = 9° on Siemens Verio 3T Scanner at Auburn University MRI Research Center. The subjects were instructed to perform no motion and...
head motion to separately acquire motionless images (reference) and real motion-corrupted images.

For the pix2pix model training, minbatch Adam solver was used with learning rate $\beta_1 = 0.5$, $\beta_2 = 0.999$. The network was trained over 200 epochs (40 hours) on an NVIDIA Tesla P100 GPU.

6.5 Results

In this section, we use the simulated test set to evaluate quantitatively the accuracy of the motion-corrected images using the proposed method. We show several examples of applying the proposed motion correction method on the simulated motion-corrupted images and real motion-corrupted images.

Three metrics—structural similarity index (SSIM), sum of absolute differences (SAD), and sum of squared difference (SSD)—were evaluated to compare the motion-corrupted images and the motion-corrected images.

SSIM is given by [96]:

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\mu_x^2 + \mu_y^2 + c_1}(\sigma_x^2 + \sigma_y^2 + c_2),$$

(6.10)

where $\mu_x$ and $\mu_y$ are the mean of the images, $\sigma_x$ and $\sigma_y$ are the variance of the image, $x$ and $y$ present two images to be compared, and $c_1$ and $c_2$ are small constants to provide calculation of stability. For SSIM, high values mean two input images are similar, but for SAD and SSD, the low values mean two input images are similar. Table 6.1 compares the average metrics between the motion-corrupted images and the motion-corrected images with the same reference images. Each entry in the table is the average value of all the 1780 cases in the simulation test set.

Results of motion correction on five simulation datasets from the simulation test set are shown in Figure 6.5. For comparing the error images (the absolute value of the difference
Table 6.1: Average values of three metrics of the motion-corrupted images and the motion-corrected images

<table>
<thead>
<tr>
<th>Images</th>
<th>SSIM</th>
<th>SAD</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{mot}$</td>
<td>0.6538</td>
<td>5.9593</td>
<td>192.2482</td>
</tr>
<tr>
<td>$I_{rec}$</td>
<td>0.6997</td>
<td>5.1133</td>
<td>134.4986</td>
</tr>
</tbody>
</table>

between the reference and input image), we use two rows for each dataset. The first row consists of three images from left to right: motionless image (reference), motion-corrected image, and error image. The second row consists of three images from left to right: motionless image (reference), motion-corrupted image, and error image. Figure 6.6 shows the results of motion correction on two experimental datasets from the experimental test set. The first column shows motion-corrupted images generated by subject motion in the experiment; the second column shows no-motion images generated by motionless subject in the experiment; the third column shows the motion-corrected images through the pix2pix model.

Figure 6.5 and Figure 6.6 show several examples of applying the proposed motion correction method on the simulated motion-corrupted images and real motion-corrupted images. The motion-corrected images indicate that our method was able to eliminate motion artifacts and to preserve the sharp boundaries. In Figure 6.5, the corrected images recovered almost all the details of the reference images. In Figure 6.6, the blurred edges became sharp and the corrected images recovered most of the details of the reference images, but there were some differences between the reference images and the corrected images.

One important note is that the error of the fifth dataset in Figure 6.5 is larger than other datasets, due to large rotation angles of the motion-corrupted image during the motion simulation. These large angles lead to the corrected image having a relative angle to the reference image. If a registration is implemented between the reference image and the corrected image, the error will be smaller. We think the relative distance between reference images and motion-corrected images has an impact on the metrics calculated in Table 6.1. The metrics can be improved by performing registration for each pair of images.
Another important note concerns the reasons for the errors in the corrected images in Figure 6.6. Two explanations may be given for this. First, the subjects’ position changed during the experiment. A small position change between two scans can lead to mismatch between motion-corrupted images and reference images. Second, information is lost due to through-plane motion. Some through-plane motion is almost inevitable when subjects performed motion, which leads to some information being lost in the reference plane and some information being obtained from other planes.
Figure 6.5: Results of motion correction on five simulation datasets from the simulation test set. For each dataset, we use two rows to compare the results. The first row consists of three images from left to right: motionless image (reference), motion-corrupted image, the absolute value of the difference between the reference and motion-corrupted image. The second row consists of three images from left to right: motionless image (reference), motion-corrected image, the absolute value of the difference between the reference and motion-corrected image.
Figure 6.6: Results of motion correction on two experimental datasets from the experimental test set. The first column shows no-motion images generated by subject motion in the experiment; the second column shows motion-corrupted images generated by motionless subject in the experiment; the third column shows the motion-corrected images through pix2pix model.
Chapter 7
Conclusions

We have developed three methods to correct MRI motion retrospectively. These three methods have their own characteristics and scope of application. The autofocus method using global motion estimation can correct rigid motion for each k-space phase line. This method required raw k-space data with the acquisition order, and the processing is relatively slow compared to the other two methods. The method based on fast robust correlation can also handle rigid motion but for each k-space segment. The main advantage of this method is that it can process the data efficiently. However, this method cannot make line-by-line correction, which may not be precise enough in some applications. Unlike the first two methods, the method using GANs does not require the raw k-space data. This method can also perform rigid motion correction efficiently, if the network has already been trained. This technique may occasionally generate images with phantom features that have no correspondence in the actual object. This is one of the biggest problems for this method, which may hinder diagnosis and scientific research. We summarize below the contributions and discoveries related to the three methods.

We developed an autofocus motion correction method using global motion estimation to address the two challenges of previous methods: high calculation load and local minima. We tried to implement some previous methods, but the motion correction results were not perfect. After a lengthy study of these methods, we found that neighboring lines in k-space have a strong relationship, and the motion detection of one segment may not correspond to the average motion of all the lines in the segment when the segment size is smaller than a certain number. Due to the strong impact of the neighboring line, the motion of the whole segment may not be corrected properly, which may result in the cost function being trapped
in local minima. This finding explained why previous methods cannot correct motion in some cases. Thus, we proposed a new search strategy. The proposed method involves two steps: piecewise linear initial approximation and joint refinement of a global model to perform global estimation. To further improve computational efficiency, golden-section search was used to estimate rotation, and two map data structures were applied to store calculated data. Different focus criteria were compared under the proposed method. Simulations and experimental results demonstrate that the proposed method can effectively and efficiently correct rigid motion (head motion) in MR images.

We also devised an efficient motion correction method based on fast robust correlation. The proposed method corrected motion-distorted images by aligning translational motion between images formed by neighboring frequency segments. To reduce the impact of the strong relationship between neighboring lines, the size of segments cannot be too small. Since the squared difference kernel was invariant to motion between partial-Fourier images, we adopted the absolute value kernel, which can be easily approximated by sinusoidal terms. Total variation of the sum of partial-Fourier images was chosen as the new matching criterion. FFTs are used to calculate correlations for computational speed. Different search strategies to combine and correct motion over the whole k-space were compared. The proposed method can perform real-time processing to reduce image motion artifacts significantly in the simulations and MRI cardiac experiments.

Finally, we designed a novel data-driven motion correction method for magnitude MR images using generative adversarial networks (GANs). GANs (pix2pix model) were implemented to reduce motion artifacts and reconstruct motion-corrupted images through adversarial training between generator and discriminator networks to estimate a motion-corrected image that should be close to the reference image. The training set was made of image pairs consisting of motionless reference images and corresponding motion-simulated images. The proposed method was validated by a simulated motion test set and a real motion (experimental) test set.
The following future work is suggested by our previous studies on this problem:

For the autofocusing method, we expect it to correct 3D rigid motion. For 3D rigid motion, six motion trajectories need to be estimated. Due to the increased search space in 3D, an accurate and efficient piecewise linear initial approximation algorithm is needed. To avoid local minima, one or more restriction conditions should be considered in the process of joint refinement of global model. The proposed method on a windowed image may potentially correct some complicated nonrigid motions. In some cases, we can assume that the whole image can be divided into several small subimages, within which the motion can be modeled as rigid. The subimages are combined after motion of each subimage is corrected by the windowed method.

For the method based on fast robust correlation, we expect it to correct translations with rotation. We may use golden-section search strategy for rotation correction. It should not be hard to extend the proposed method to correct 3D translations, which will make the method more powerful. The proposed method applied to a windowed image will be another attractive extension.

For the method using GANs, we expect it to correct some simple non-rigid motions, such as some motions of joints. Instead of using the same networks for all the motions, a specific network structure and objective function is needed to better fit the motion properties of a certain part of the body. In order to compare different network methods, an authoritative MRI dataset and some metrics should be established.
Bibliography


