Simultaneous Registration and Activation Detection: Overcoming Activation-Induced Registration Errors in Functional MRI

by

Jeffery J. Orchard

B.Math., University of Waterloo, 1994

M.Sc., University of British Columbia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science

SIMON FRASER UNIVERSITY

April 2003

APPROVAL

Name: Jeffery J. Orchard

Degree: Doctor of Philosophy

Title of thesis: Simultaneous Registration and Activation Detection: Overcoming Activation-Induced Registration Errors in Functional MRI

Examining Committee: Dr. Richard (Hao) Zhang
Chair

Dr. M. Stella Atkins, Senior Supervisor

Dr. Torsten Möller, Supervisor

Dr. Bruce Bjornson, Supervisor
Professor of Pediatrics
University of British Columbia, Vancouver

Dr. Bob Russell, Professor of Mathematics
SFU Examiner

Dr. Michael Fitzpatrick, External Examiner
Professor of Computer Science
Vanderbilt University

Date Approved: _____

Abstract

In the processing of functional magnetic resonance imaging (fMRI) data, motion correction is typically performed before activation detection. However, on high-field MR scanners (3 T and higher), the strength of the blood oxygen level dependent (BOLD) signal can cause registration algorithms to produce motion estimates that have stimulus-correlated errors. Motion compensation using these biased motion estimates can result in both false-positive and false-negative regions of activation.

By formulating the registration and activation detection problems into a single least-squares problem, both the motion estimates and activation map can be solved for simultaneously. However, the solution is not unique and an additional constraint is used to find a solution that is appropriate. This constrained optimization problem can be solved efficiently, and two equivalent methods are proposed and demonstrated on both simulated and *in vivo* datasets.

In loving memory of my brother Michael,
whose spirit rains clairty onto the hubbub of my day-to-day life.

Acknowledgments

I would like to thank L. M. Freire, J.-F. Mangin, S. Berthoz, P. F. Van de Moortele, J. L. Martinot, D. Le Bihan, and the Service Hospitalier Frédéric Joliot (SHFJ) for supplying me with *in vivo* datasets I, II and III, as well as F. G. C. Hoogenraad of Philips Medical Systems for supplying me with *in vivo* dataset IV.

In addition, I would like to thank Chen Greif and Gene Golub for their mathematical insights, as well as their advice, which I am glad to say I had the wisdom to follow.

To my senior supervisor, M. Stella Atkins, I owe a heap of gratitude. Her guidance in every aspect of my scholarly life has been invaluable. She is truely my academic mentor.

Finally, I thank my wife, Tricia, for supporting and encouraging me throughout my journey. Getting up early to see her off to work kept my study habits in check.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

While many medical imaging modalities are used to investigate tissue anatomy, some are designed to observe tissue function. In this context, the word "function" refers to activity in the tissue, as opposed to the tissue itself. For example, positron emission tomography (PET) and single photon emission computed tomography (SPECT) both sense the decay events of radio isotopes to view activity in the brain and other organs. While PET and SPECT each have a wide range of uses, the fact that the patient sustains a dose of radiation creates the demand for alternative functional imaging methods, especially for large scientific studies involving volunteer subjects.

Magnetic resonance imaging (MRI), an imaging modality that has no known side-effects[1], can be used to gain information about anatomy, blood perfusion, chemical composition, water diffusion, and many other physiological characteristics. The uses of MRI even include functional imaging of the brain, where an MR scanner can sense minute blood-flow changes correlated to neurological activity.

1.1 Notational Conventions

Before we continue, a few notational and naming conventions will help to keep many of the concepts, and their relationships to each other, more organized. First of all, medical imaging data comes in different dimensions. In general, 1-dimensional (1D) data is often called a "function", while a 2D array of data is called an "image", and a 3D array of data is called a

[1] An MR scanner can have ill effects when the patient has metal embedded in soft tissue, or has life-supporting electronics such as a pacemaker, but is otherwise considered to be harmless.

"volume". However, there are situations where the dimension of the data is not important. In particular, the words "image" and "volume" are sometimes used interchangeably. In this thesis, a common form of data is a time series: an ordered series of associated images acquired in rapid succession. This data, which has an inherent time dimension, will be referred to as a "dataset". Note that a dataset can be a time series of 2D images, or 3D volumes (here we have already used the word "image" to represent both 2D and 3D data). The action of acquiring a dataset (time series of images) is called an "experiment". The phrases "imaging sequence", "scanning sequence" and "acquisition" are more general terms used to refer to the act of acquiring imaging data from the scanner. The term "pixel" refers to one element of an image, while "voxel" refers to one element of a volume. When the dimension of the dataset is ambiguous, we use the word "sample" instead.

Some algebraic conventions are also worth mentioning at this point. There are two common notations for vector variables. Here we use both conventions, but for two different types of vectors: column-vectors, and row-vectors. Spatial data, such as an image or volume, will be stored in a column-vector and denoted using a lower-case bold letter, such as $\mathbf{f}$. Time-series data will be stored in a row-vector and denoted using a vector-arrow over a lower-case italic letter, such as $\vec{f}$.

Since a column-vector is 1D, storing an image (or volume) in a column-vector requires that the pixels (or voxels) be ordered into a sequence. The particular ordering does not matter, as long as a consistent ordering is used throughout. This 1D sequencing allows individual samples (pixels or voxels) to be denoted using a single subscript, such as in $\mathbf{f}_n$. We may also denote a single time-series sample as $\vec{f}_n$. If $\mathbf{f}$ represents a 2D image, then a single pixel can be specified using two subscripts given in parentheses, as in $\mathbf{f}_{(m,n)}$. In some cases, it is more convenient to place the indices in square brackets, such as $\mathbf{f}[m, n]$, rather than as subscripts.

The reason for putting time series in rows, and images (or volumes) in columns is to allow a time series of images to be stored in a single matrix. Figure 1.1 shows a schematic of this matrix design. Hence, for a dataset containing n time-series snapshots of m voxels each, we define the *dataset matrix* $\mathbf{F}$ with n columns and m rows.

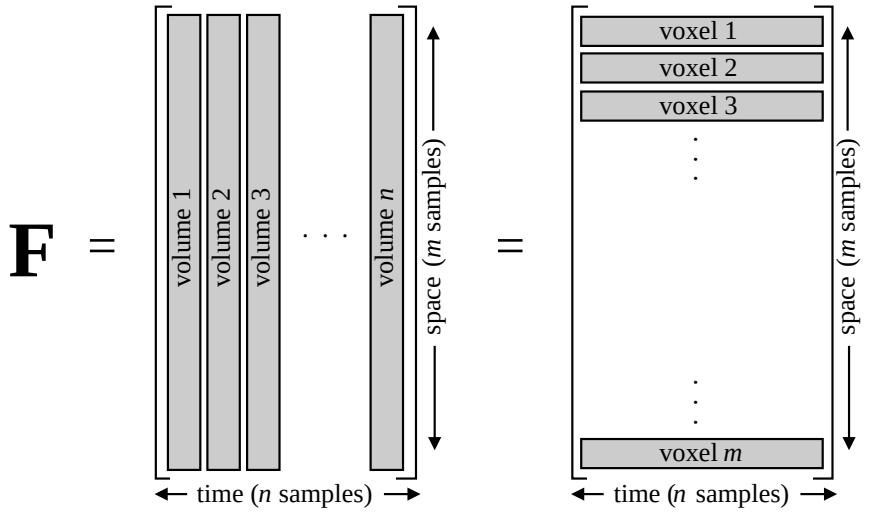


Figure 1.1: Dataset matrix design. An entire time series of volumes can be stored in a single matrix, with volumes stored in columns, and voxel time series stored in rows.

1.2 BOLD Signal

Functional magnetic resonance imaging (fMRI) takes advantage of a phenomenon known as the blood oxygen level dependent (BOLD) contrast. Hemoglobin, the molecule in blood that binds with oxygen and carbon-dioxide, is paramagnetic. Paramagnetic materials tend to align with the magnetic field, and hence can be sensed by the MRI scanner. Oxygen-rich blood has a brighter MR signal than deoxygenated blood [60].

Consider a small area of brain tissue that briefly undergoes increased activity. Shortly after the start of the activity, the tissue's use of glucose increases, causing a compensatory influx of blood rich in glucose and oxygen. The MR signal increases due to the high concentration of oxyhemoglobin. After a short time (typically 10 to 20 seconds), the blood flow stabilizes back to its normal equilibrium, and the MR signal returns to its nominal value. This transient, shown in Fig. 1.2, is referred to as the *BOLD signal*, or *hemodynamic response*. An MRI acquisition taken over a 20-second period can detect the BOLD signal changes that occur in active regions of the brain.

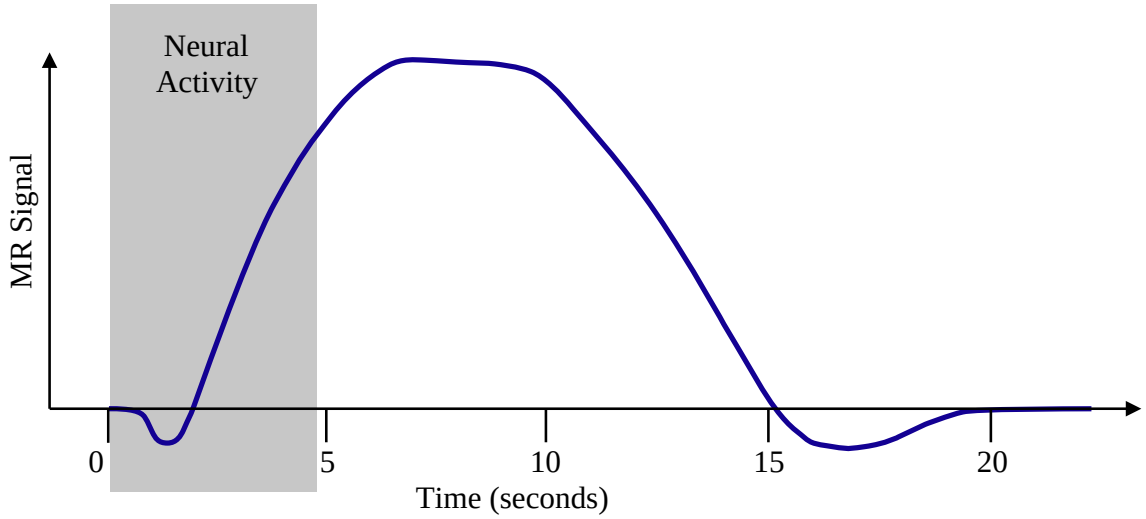


Figure 1.2: BOLD Signal for a single, brief neurological stimulus. The graph shows the evolution of the MR signal typically exhibited by active brain tissue over approximately 20 seconds. This curve is also known as a hemodynamic response function.

1.3 Statistical Correlation

The goal of fMRI studies is to establish a *statistical parametric mapping*, assigning every voxel to a class according to its statistical properties [34] (recall that a voxel is a sample of a 3D function, and is the 3D analog of a 2D pixel). Snapshots of the subject's brain are acquired repeatedly using a special high-speed MR scanning sequence that is sensitive to the BOLD signal. During the experiment, the subject is presented with carefully chosen neurological stimuli. The objective is to discover voxels with variations in intensity that follow the same pattern as the stimulus. For example, the subject may be shown a flashing pattern designed to stimulate the left side of their visual field. This pattern can then be changed to stimulate the right side of the visual field. The two different stimuli are designed to activate different parts of the brain. The reasoning behind this design is that the localization of activation in the brain depends on the contrast between the two states presented to the subject. Due to the BOLD contrast, tissue contained in a voxel that is involved in the neurological processing of one of the stimuli, but not the other, will show up bright in one state, but dark in the other.

The different states created by the presentation of different stimuli are called *conditions*. For example, a flashing pattern in the left visual hemifield might be referred to as condition 1,

while the right visual hemifield pattern presentation might be called condition 2. Or, in a different experiment, condition 1 might require the subject to listen to a sound and press a button if the sound is a steady tone, while condition 2 might involve the subject watching moving dots. It is important to point out that the lack of a presented stimulus is also considered to be a condition. If an experiment involves a visual task, interleaved by periods of rest, then condition 1 is the task, and condition 2 is rest. Each condition elicits its own neurological response that is (hopefully) discernible from that of the others.

This switching of stimuli is executed many times so that contrast can be determined with statistical significance. A typical fMRI experiment may have 10 cycles, each cycle consisting of one stimulus being presented for a time, and then the other. Each cycle usually lasts between 10 and 30 seconds: long enough for the BOLD response to materialize, but short enough that the cycle can be repeated often. The result is a time series of MRI volumes. Thus, for each voxel there is a time series of intensity values, one from each volume.

For simplicity, we will assume that the subject did not move during the experiment. Issues of subject motion will be addressed in section 2.3.

Suppose we know of a particular voxel that contains brain tissue that is activated by condition 1, but not by condition 2. We would expect its time series to be related to the presentation of the stimulus. Furthermore, if that voxel of tissue has the hemodynamic response shown in Fig. 1.2, we would expect the observed signal from that voxel to be a combination of its hemodynamic response and the time series of the stimulus presentation. If we assume that the stimulus-response system is linear and time-invariant, then we can use the hemodynamic response kernel convolved with the stimulus train to derive the expected hemodynamic response. Figure 1.3 shows a stimulus function and the expected hemodynamic response function generated by convolving the stimulus function with the BOLD signal in Fig. 1.2.

The challenge of fMRI activation detection is to determine which voxels have time series that closely match the stimulus. These voxels indicate which areas of the brain are activated by the stimulus.

Since the BOLD signal is very small, several volumes are added together and averaged to increase the signal-to-noise ratio. Typical experiments last for 3 to 10 minutes, during which time the patient might move. Therefore, consecutive volumes in a dataset must be registered.

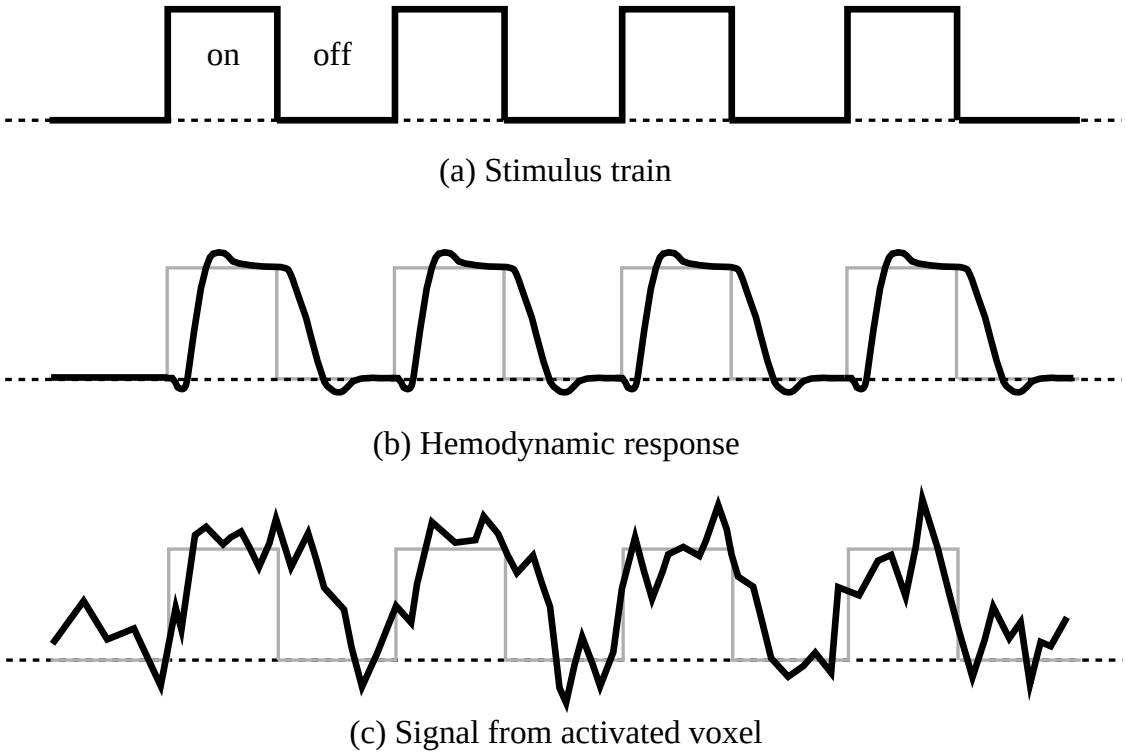


Figure 1.3: By convolving the BOLD response curve in Fig. 1.2 with the stimulus train shown in (a), the result is the expected hemodynamic response function, shown in (b). The time series response from a typical active voxel is shown in (c).

1.4 Patient Motion and Activation Bias

Patient motion can have a profound, negative impact on the accuracy of fMRI results. The mis-alignment of volumes can artificially increase the variation of a voxel's time course, and prevent the detection of the BOLD signal [35, 83]. The resulting activation maps are often subject to false-negatives, the mis-classification of active brain regions as inactive.

A common approach to overcome patient motion is to automatically estimate the motion and use the estimates to correct the dataset. This process of aligning the volumes is called *registration*. After the registration algorithm aligns the volumes in the dataset, the resulting activation maps are more accurate [83]. One of the most widely-used motion detection methods is least-squares registration. This method attempts to minimize the sum of the squared residuals between the two volumes being compared. It is computationally efficient, and the sum of squares cost function has been shown to be optimal when the two volumes differ only by additive Gaussian noise [44].

However, it has been shown [30, 31] that the BOLD signal present in fMRI datasets acquired on high field scanners (3 T and higher) may violate the assumptions underlying many registration methods. The unit "T" stands for "Tesla", and is a unit of magnetic field strength (see appendix A for details on MR scanner magnetic fields). When two volumes are being compared, the presence of BOLD contrast in one image, but not the other, causes the corresponding regions to be mis-matched. The impact of this phenomenon was reported by Freire *et al.* [30]. In this paper, Freire refers to these mis-matched regions as "outliers" in the statistical fit. It is well-known that the sum of squares cost function, in general, is particularly sensitive to outliers. Using this cost function for registration in fMRI yields motion errors that are highly correlated to the stimulus function. When the resulting inaccurate motion estimates are used in the motion "correction" process, some of the actual motion is removed, but erroneous motion is introduced. Since the false motion is caused by the brain activation, the false motion is correlated to the stimulus. Thus, the resulting dataset has stimulus-correlated motion artifacts.

Patient motion has a particularly insidious effect when it is synchronized with the stimulus. This type of motion causes voxel intensity fluctuations that mimic those of active voxels. As a result, inactive voxels can take on the appearance of being active [35]. A comprehensive study of the effects of stimulus-correlated motion on fMRI activation maps was conducted by Hajnal *et al.* [41]. During their study, seven of the eight *in vivo* datasets

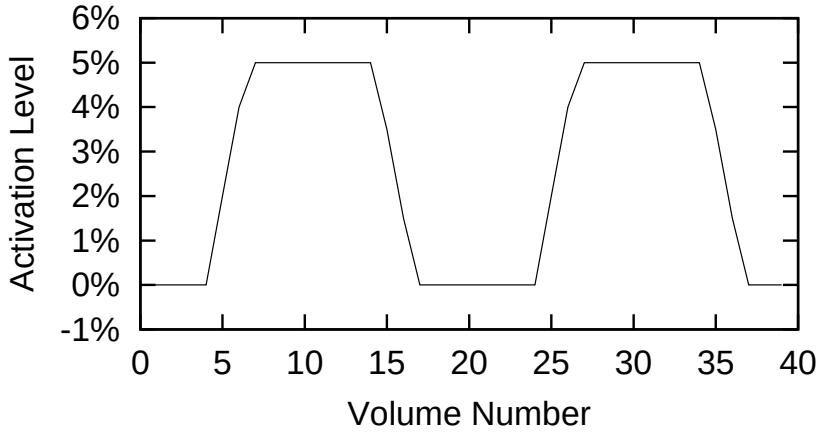


Figure 1.4: Stimulus function for simulated datasets.

they acquired contained significant stimulus-correlated motion. They also showed that the motion extracted from these datasets was capable of producing the observed activation even without the added activation. The study illustrates that motion cannot be assumed to be independent of the stimulation paradigm, and that stimulus-correlated motion has an overwhelming potential to produce false-positive activations. The stimulus-correlated errors present in the motion parameter estimates of least-squares registration techniques are problematic because they introduce stimulus-correlated motion into the dataset.

Freire reported [30] that, on datasets with voxel dimensions $3.75 \times 3.75 \times 4$ mm (which is typical for fMRI), motion errors as small as 0.05° and 0.05 mm resulted in the detection of false-positives. Freire's findings have also been supported by a theoretical study of the relationship between activation and least-squares motion errors [76].

Similar results have been reproduced here by running experiments on simulated fMRI datasets. Our datasets were designed to be comparable to those of Freire. Simulated fMRI datasets were created with activation according to the stimulus function shown in Fig. 1.4 (for a full description of how the simulated datasets were created, see section 4.1).

Figure 1.5 plots the motion estimates resulting from a least-squares registration method on a motion-free simulated dataset. The Figure contains six graphs corresponding to the six motion parameters of 3D, rigid-body motion: three translation parameters, and three rotation parameters. The translations are along the three scanner-fixed coordinate axes, which

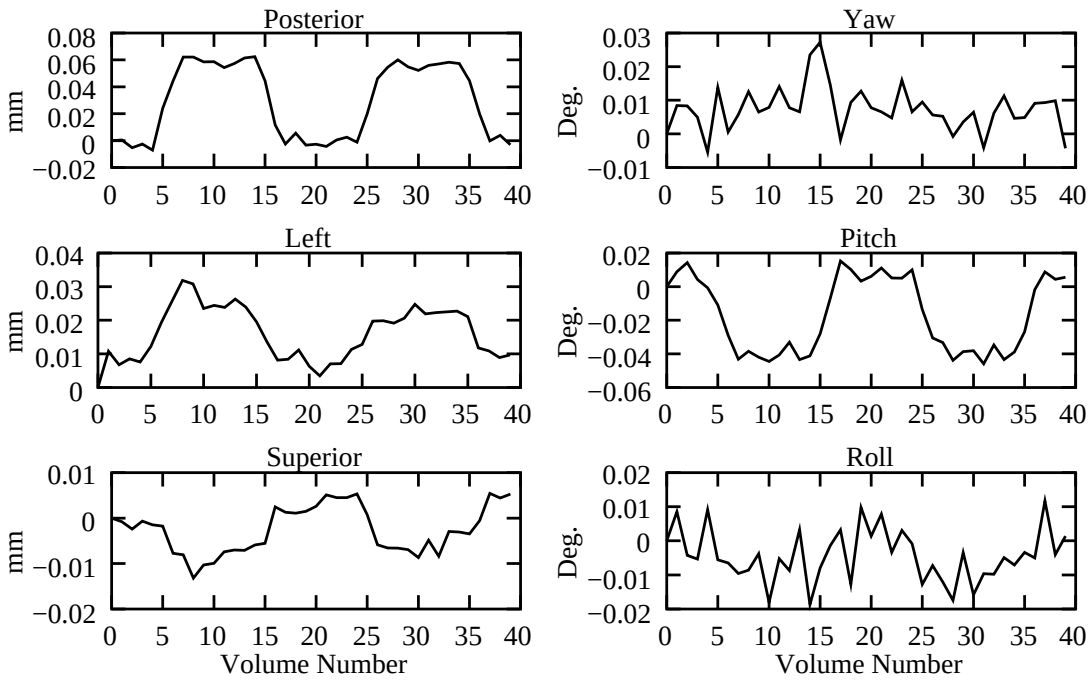


Figure 1.5: Least-squares motion parameter estimates for a typical motion-free simulated dataset. Notice the high correlation between some of the motion parameters and the stimulus function shown in Fig. 1.4. The graph labeled "Posterior" shows the error in the anterior-to-posterior translation motion parameter. Similarly, "Left" stands for right-to-left translation, and "Superior" stands for inferior-to-superior translation. The yaw angle is the angle of rotation about the anterior-posterior axis, while the pitch angle is measured about the left-right axis, and the roll angle is measured about the inferior-superior axis.

are defined with respect to the anatomy of the patient laying in the scanner: the posterior-to-anterior (back-to-front) axis, the left-to-right axis, and the superior-to-inferior (top-to-bottom) axis. The rotations are defined as follows: yaw is rotation about the posterior-to-anterior axis, pitch is rotation about the left-to-right axis, and roll is rotation about the superior-to-inferior axis. Each graph exhibits the changes of a motion parameter throughout the course of the fMRI experiment. Since the dataset actually has zero motion, any non-zero motion estimates are erroneous. In this sense, the plots in Fig. 1.5 show the errors in the motion estimates. A registration algorithm that is robust to activation should have error plots that hover around the zero mark. The graphs shown in Fig. 1.5, however, exhibit large displacements from zero during the periods of stimulation.

1.5 Thesis

It is clear that motion should be removed from the dataset before activation detection is performed. But it also seems that activation should be removed from the dataset before registration is done. It stands to reason that these two processing steps should be considered coupled, and neither should be solved before the other. By combining the effects of both motion and activation into a single mathematical model, the two problems may be solved simultaneously. A simultaneous solution allows the registration problem to take the activation into account, and allows the activation detection problem to take the motion into account.

In chapter 2, the background of fMRI and motion correction techniques is presented. Then, in chapter 3, we formulate the registration and activation detection problems into a single least-squares problem, and solve for both motion and activation simultaneously. Finally, results of testing this new simultaneous solution are presented in chapter 4.

Chapter 2

Background

2.1 Magnetic Resonance Imaging Basics

In this section, we outline some of the basic concepts behind MR imaging. For a much more detailed discussion on the topic of MR physics, see appendix A.

Protons in the tissues of the body have a property called the *dipole spin*. These "dipoles" have a tendency to align themselves with a strong magnetic field. Associated with each dipole is a frequency called the *Larmour frequency* that depends on the local magnetic field strength. Thus, in the presence of a magnetic field gradient, a particular Larmour frequency corresponds to a plane of tissue perpendicular to that gradient. Applying a radio frequency (RF) pulse at that frequency causes the dipoles in that plane to resonate. This is called *slice excitation*. Once a slice is excited, the combined effect of all the resonating dipoles creates a small magnetic field that can be sensed by the MR scanner. The vector that represents the direction and magnitude of this magnetic field is called the *net magnetization vector*.

The dipoles in different tissues behave differently, depending on the composition of their micro-environment. These differences are exhibited in the local tissue's net magnetization vector and are typically characterized by three tissue parameters denoted T_1, T_2 and PD. The parameters T_1 and T_2 are decay rates that determine how quickly the net magnetization vector returns to normal after excitation, while PD stands for proton density, and is a measure of just that. The signal differences measured by the MR scanner are caused by tissues with different T_1, T_2 and PD values.

A scanning sequence usually consists of collecting multiple 2D slices of data one slice

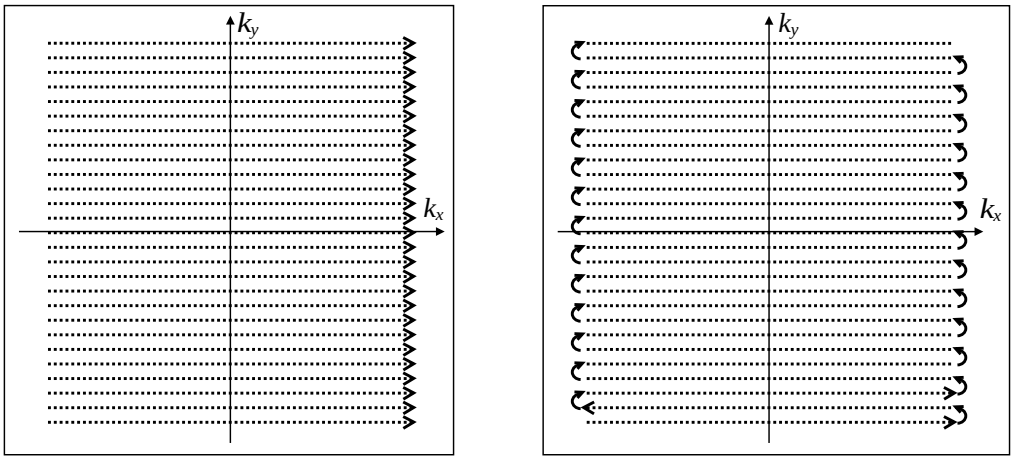
at a time. However, in MR imaging, one cannot collect image samples in a spatially-ordered manner. What is actually measured by the MR scanner is the sum of all the net magnetization vectors in the slice. Thus, each individual measurement gives information about every voxel simultaneously. Once all the measurements are acquired for a 2D slice, the mass of data can be decoded to derive a location-specific intensity value for each voxel.

The acquisition of each 2D image slice starts by exciting the tissue in that slice using an RF pulse in the presence of a magnetic field gradient. The objective of the imaging process is to sample the frequency and phase characteristics of the slice of tissue sufficiently to derive the intensity at each voxel. The specific information gathered by each measurement is differentiated by controlling the degree to which the dipoles are out of phase. The dipole phases can be manipulated using two additional magnetic field gradients in the scanner. The action of these gradients is quantified by a pair of k variables. Changing the values of these k's enables the scanner to measure different information in each sample. The 2D space of k-values is called "k-space". A common method to acquire all the necessary samples in k-space is to collect samples on a regular (uniform) grid.

These k-space samples are actually the frequency domain representation of the image data, so the image itself can be attained by simply taking the inverse Fourier transform of the k-space grid of data. Properties of the Fourier transform are discussed in appendix B.

In anatomical MR imaging, typically only a single line of this grid is acquired during each RF excitation. It takes many RF excitations to acquire an entire slice of data. This method, called *spin-echo imaging*, is not fast but it creates images with high resolution and relatively little noise. Figure 2.1(a) shows the spin-echo k-space traversal.

For fMRI, a faster k-space traversal technique called *echo-planar imaging* (EPI) is used. In this method, the whole 2D grid of samples is acquired after a single RF excitation, as shown in Fig. 2.1(b). Instead of stopping after the first k-space line (like in spin-echo imaging), the traversal turns around and samples the next k-space line in the opposite direction, followed by the next line, and so on. This back-and-forth traversal happens very quickly, but has its drawbacks. Because of the speed, this method produces images with lower resolution and more noise than the spin-echo method. Also, EPI suffers from certain artifacts that the spin-echo scanning sequence suppresses. However, the EPI scanning sequence brings out particular tissue contrasts that are necessary in functional MR imaging.



(a) Spin-Echo (b) EPI

Figure 2.1: Two common k-space traversal strategies. The spin-echo imaging traversal is depicted in (a), and the echo-planar imaging traversal is depicted in (b).

2.2 Functional Magnetic Resonance Imaging

2.2.1 Blood Oxygen Level Dependent Signal

The property on which functional MR imaging is based is called the *blood oxygen level dependent*, or BOLD, signal. As the name implies, the signal that the scanner receives is related to the amount of oxygen in the blood. Just how this happens is a slightly involved story.

Consider a small volume of brain tissue that briefly undergoes increased neurological activity. The recycling of the neurotransmitters is an anaerobic process and requires glucose, but little oxygen. A small local increase in blood flow delivers the needed glucose to the tissue. However, this blood is also rich in oxygen. This inflow of oxygen, coupled with the fact that little oxygen is used in the recycling of the neurotransmitters, causes the local concentration of oxyhemoglobin to increase.

Hemoglobin, the molecule in blood that binds with oxygen and carbon-dioxide, is paramagnetic. Paramagnetic materials tend to align with the magnetic field, and hence can be sensed by the scanner. An increase in deoxyhemoglobin increases the local T_2 decay. The parameter T_2, known as the *transverse relaxation time* or *spin-spin relaxation time*, is a

characteristic of a tissue's composition. Magnetic resonance scan sequences are tuned to take advantage of these tissue differences and create images that exhibit strong contrast between tissue types (for information about MRI physics and T_2 decay, see appendix A.1).

Conversely, an increase in oxyhemoglobin decreases the local spin variation, causing a slower T_2 decay. As a result, tissue rich in deoxyhemoglobin has a lower T_2 signal, while tissue rich in oxyhemoglobin has a higher T_2 signal. The signal picked up in a T_2-weighted acquisition of an active region of the brain is higher than that of an inactive region because the active region has a higher concentration of oxyhemoglobin. After a short time (typically 10 to 20 seconds), the blood flow stabilizes back to its normal equilibrium, and the signal returns to its normal value. This transient, shown previously in Fig. 1.2, is referred to as the *BOLD signal*, or *hemodynamic response*.

It should be noted that the specific shape of the BOLD signal curve can vary from person to person and from brain region to brain region.

2.2.2 Physiological Motion

The patient's breathing and heart beat can cause signal fluctuations in fMRI data large enough to reduce the significance of the activation detection results. The specific mechanism by which breathing and heart beat affect the images has not been demonstrated conclusively. However, substantial evidence suggests that the changing composition of the chest cavity is the most likely culprit, since motion of tissue alters the scanner's magnetic field [82, 104].

It is important to design an fMRI experiment so that the frequency of the stimulus does not correspond to the frequency of any other factors influencing the data. The correction methods below depend on this constraint.

Band-block filtering has been used to remove physiological effects by discarding signal components that fall within a specified range of frequencies. However, the method assumes that the physiological signal is periodic. Respiratory and cardiac cycles are pseudo-periodic: the cycles repeat with a slightly random period. This is not the same as true periodic motion, and the resulting signal's frequency spectrum may overlap that of the hemodynamic function causing a band-block operation to either allow some physiological variation to remain, or forcing some BOLD signal to be removed. This is not desirable.

Triggering techniques have been used to combat the effects of physiological motion. The physiological signal is monitored separately, and the image acquisition is started at the appropriate point in the physiological state. However, triggering cannot be applied to fMRI

because the EPI scan sequence depends on the regular timing of the image acquisitions (see appendix A.5 for information on the EPI scanning sequences).

In gating, the excitation cycle runs continuously, but data is only acquired if the acquisition time coincides with the appropriate physiological state [52]. Gating is also not optimal for fMRI because it tends to lengthen the acquisition time considerably. Subject attention may wane after a few minutes of task. Breath holding can eliminate the respiratory artifacts for up to one minute, but fMRI experiments often take longer than three minutes. Naturally, breath holding does not change the effects caused by cardiac pulsation.

One approach designed to correct intra-scan physiological motion artifacts was demonstrated by Hu *et al.* [47]. A navigator echo was used to collect a single k-space point prior to the acquisition of each k-space line (see section 2.3.2 for a description of navigator echoes). This phase measurement was used to correct the phase of the samples accordingly. Hu *et al.* demonstrated that cancellation of the physiological motion artifacts reduces the time-series variation significantly, and allows greater sensitivity of activation detection for fMRI experiments. This method corrects for physiological artifacts within an image. It does not correct for such artifacts between images.

Physiological fluctuations between images in a time series can also be corrected. The respiratory cycle can be monitored during the imaging sequence by a flexible pressure belt. Using this information about the patient's breathing, the scanning period can be broken into breaths, each separated by the point of maximum inhalation. The samples collected during all the breaths are superimposed on a single unit-cycle graph, such as that shown in Fig. 2.2. That is, a sample that is taken 2 seconds after the beginning of an 8-second breath is assigned an offset of 0.25, meaning it was collected one-quarter the way through the breath. The resulting scatter plot is then least-squares fitted with a truncated Fourier series. Figure 2.2 shows the scatter plots with the corresponding fitted curves. The curve fit indicates the average effect due to respiration over the cycle. This average physiological variation is subtracted from the dataset samples. For every sample, the phase and magnitude are shifted to remove the effect of the physiological fluctuation. This correction procedure is then followed by the same procedure to correct for cardiac pulsations [48].

The advantage of this technique is that it allows for variation in the period of the pseudo-periodic events. For instance, a subject may breathe approximately once every 5 seconds. However, some breaths may be 10 seconds long, while others only 3 seconds. The independently-measured physiological data records these fluctuations and the MR data is

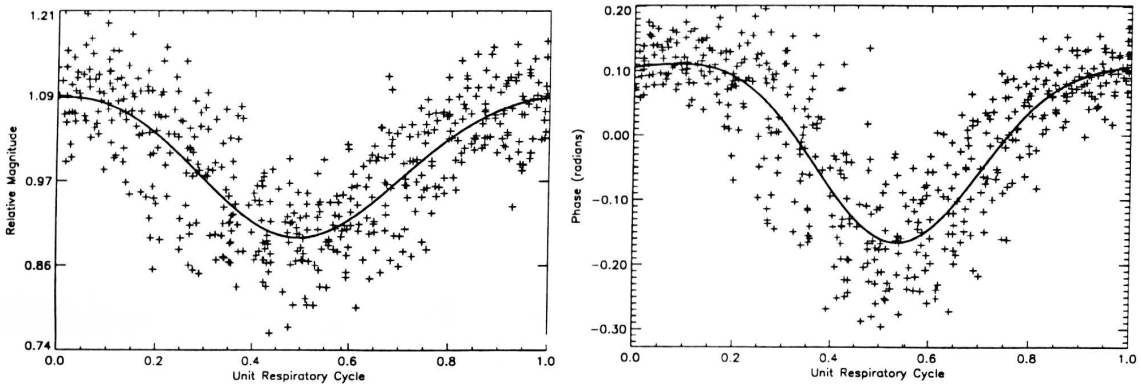


Figure 2.2: Magnitude (left) and phase (right) of samples in the unit respiratory cycle. The solid lines represent the truncated Fourier-series fit to the samples. (Taken with permission from Hu *et al.* [48])

corrected accordingly. Figure 2.3 plots the time series for an activated voxel during an fMRI experiment, before and after the application of this correction procedure.

One disadvantage to this method is that it cannot be applied to datasets for which respiratory and cardiac monitoring was not done. Le and Hu extended this technique by eliminating the need for physiological monitoring in some cases [54]. Instead, if the image data was acquired at a sufficiently high frequency, then the data itself can be used to derive the physiological fluctuations. For repeated EPI acquisitions of a single slice, a full-slice snapshot can be acquired every 50 ms. The period of a heart beat is between 330 ms and 1500 ms. Clearly, the sampling rate satisfies the Nyquist criterion [75], allowing the cardiac cycle to be represented accurately.

The methods above apply the correction to the k-space data. Similar methods have been demonstrated to work in the spatial domain. One such method [18] simply uses the time series of the spatial voxels the same way as Le and Hu. The Fourier transforms of the voxel time series are calculated, and the most common peak frequency in the power spectra is chosen as the basic frequency of the corresponding physiological fluctuation. A voxel (or group of voxels) with a peak frequency close to this basic frequency is chosen and a filtered version of its time series is used to represent the time course of the physiological signal. A version that uses independent monitoring of the respiratory and cardiac cycles has also been implemented for correction in the spatial domain [38].

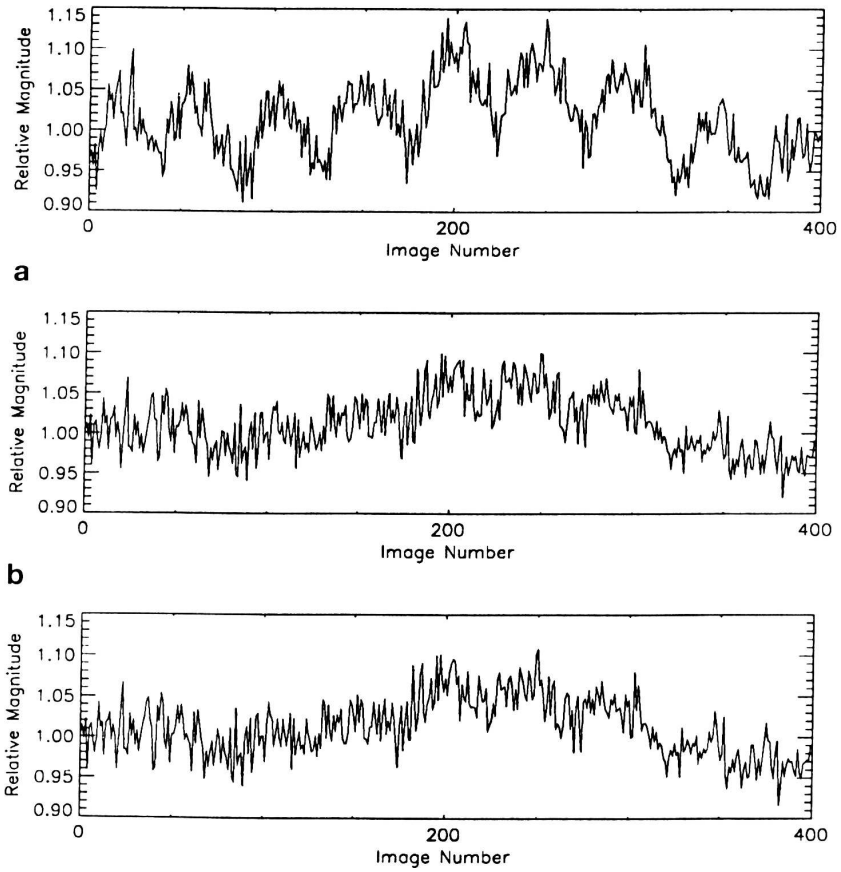


Figure 2.3: Time series of a voxel that is exhibiting substantial physiological fluctuation. The three graphs, from top to bottom, show a time series before correction, after correction using physiological monitoring, and after correction using only the data itself (the physiological signals were estimated directly from the data). (Taken with permission from Le and Hu [54])

2.2.3 Activation Detection

The goal of activation detection is to assign, to each and every voxel, a value that represents the degree to which the contained tissue is activated by a particular stimulus. The result is a volume of such values, called an *activation map* or *statistical parametric map*. In this section, we examine different techniques to measure this similarity and produce activation maps.

Correlation Coefficient

One of the first attempts to derive an activation component from the BOLD signal was done by Bandettini *et al.* [9]. The most simplistic of these methods is image subtraction. Two EPI volumes are needed: one during condition 1, and one during condition 2. Brain regions that respond differently to the different conditions will be bright in one image and dark in the other image. By subtracting the volumes, these regions of activation are highlighted. The problem with this method is that it is susceptible to artifacts. High-intensity regions in the difference image are often the result of spurious blood flow effects, CSF flow, or noise.

By averaging over a number of volumes, the noise effect can be reduced. Let the time series for a particular voxel intensity be denoted by $\vec{f}$, where $\vec{f}_i$ represents the i^{th} value in the time series. Let the vector $\vec{r}$ be a time series that represents the state of the stimulus such that $\vec{r}_i = 1$ if condition 1 is presented during the acquisition of volume i, and $\vec{r}_i = -1$ otherwise. Thus, $\vec{r}$ is a series of ones and negative ones corresponding to the stimulus train (see figure 1.3). The averaged image subtraction result can be calculated using $\sum_i \vec{f}_i \vec{r}_i$. This expression is equivalent to the vector dot-product $\vec{f} \cdot \vec{r}$. The value returned is the same as calculating an average value for voxel intensities corresponding to condition 1, and subtracting from that an average voxel intensity corresponding to condition 2.

A measure of the degree to which $\vec{f}$ and $\vec{r}$ are linearly related can be quantified by the *correlation coefficient*. The definition of the Pearson product-moment cross correlation is

$$\rho(\vec{f}, \vec{r}) = \frac{\sum_i (\vec{f}_i - \mu_{\vec{f}})(\vec{r}_i - \mu_{\vec{r}})}{\left[\sum_i (\vec{f}_i - \mu_{\vec{f}})^2 \sum_i (\vec{r}_i - \mu_{\vec{r}})^2\right]^{\frac{1}{2}}} \ , \tag{2.1}$$

in which $\mu_{\vec{f}}$ and $\mu_{\vec{r}}$ are the average values (over time) of $\vec{f}$ and $\vec{r}$, respectively. If we create new vectors $\vec{\mu}_{\vec{f}} = [\mu_{\vec{f}}, \mu_{\vec{f}}, \ \ldots \ , \mu_{\vec{f}}]'$ (where the apostrophe represents the matrix transpose operator) and $\vec{\mu}_{\vec{r}} = [\mu_{\vec{r}}, \mu_{\vec{r}}, \ \ldots \ , \mu_{\vec{r}}]'$ as simple repetitions of the corresponding

average values, then equation (2.1) can also be formulated using vector notation. Letting $\vec{F} = \vec{f} - \vec{\mu}_{\vec{f}}$ and $\vec{R} = \vec{r} - \vec{\mu}_{\vec{r}}$, the expression for correlation is

$$\rho(\vec{F}, \vec{R}) = \frac{\vec{F} \cdot \vec{R}}{\|\vec{F}\| \, \|\vec{R}\|} \ . \tag{2.2}$$

This well-established statistical measure varies in value between -1 and 1. A value of 1 indicates that $\vec{F}$ and $\vec{R}$ are positively correlated, signifying that as the value of elements in $\vec{F}$ increase, the elements of $\vec{R}$ also increase by the same factor. A correlation value of -1 (negative correlation) means the opposite: as one increases, the other decreases. A value of 0 means that there is no linear relationship between the two at all. The correlation coefficient was first used in fMRI activation detection by Bandettini *et al.* [8].

Other sources of signal variation may be present and decrease the accuracy of the correlation measure. For example, it is often desirable to remove any temporal linear trends in the data. Such trends can be the result of a slowly changing gradient coil temperature. A method is needed to decompose the voxel time series into these different components.

General Linear Model

Determining whether or not a voxel time series appears to be influenced by the stimulus can also be posed as a least-squares problem using the general linear model (GLM) [34, 53]. The idea behind the GLM is that a voxel time series, $\vec{f}$, is the sum of effects from different sources. For example, not only does each voxel have a constant, baseline intensity, but it may also have a component that increases over time, or a component that fluctuates with the stimulus. These components are called "regressors". The GLM is based on the idea that some linear combination of the regressors produces a time series that is close to the voxel's actual time series. Activation detection consists of decomposing a voxel's time series by finding a linear fit coefficient, or multiplier, for each of these regressors.

Recall that for notational consistency, time-series data is stored in row-vectors, and row-vectors are denoted using the vector arrow notation, $\vec{f}$. Also, spatial data such as images or volumes is stored in column-vectors, and is denoted using bold, as in $\mathbf{f}$.

Suppose we want to decompose a voxel's time series into s components. Then, for every voxel time series, $\vec{f}$, there is a set of s parameters

$$\vec{y} = [y_1, y_2, \ \ldots \ , y_s] \ , \tag{2.3}$$

associated with the set of s regressor vectors

$$\mathbf{B} = \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vdots \\ \vec{b}_s \end{bmatrix} , \tag{2.4}$$

such that

$$\vec{f} \approx y_1\vec{b}_1 + y_2\vec{b}_2 + \cdots + y_s\vec{b}_s \tag{2.5}$$

$$\vec{f} \approx \vec{y}\,\mathbf{B} . \tag{2.6}$$

The rows of $\mathbf{B}$ hold regressors, linearly independent vectors chosen specifically because they are thought to be components of $\vec{f}$. For example, if $\vec{b}_1$ is a row-vector of ones, then a bias can be added to $\vec{f}$ by changing y_1. That is, the same amount can be added to each element of $\vec{f}$ by letting y_1 be non-zero. This constant baseline shift must be included in fMRI time-series analysis. Another common component that is modeled is linear drift in time. The voxel intensities may slowly increase during the experiment. This drift is sometimes attributed to a small temperature change in the scanner's super-cooled magnetic coils, which affects the "static" magnetic field. Linear drift in time can be incorporated into the model by including the vector $[0, 1, 2, \ldots, n-1]$ as a regressor. If $\vec{b}_2$ is such a linearly increasing function (row-vector), then the slope of the drift is y_2.

In fMRI analysis, the expected hemodynamic response function is included as one of the regressors, and is called a *stimulus regressor*. Then, one parameter (one of the y's) gives the coefficient of the stimulus-related component of the time series. Delayed versions of the hemodynamic response can also be included, giving the decomposition more choices for the fit. In fact, even different types of hemodynamic responses can be added. The regression fitting will choose which one best fits the data by assigning a large value to the corresponding regression parameter. In this sense, the method is very flexible. However, one must be careful when including multiple stimulus regressors because it can result in overfitting, the ability for even inactive voxels to be modeled as a combination of stimulus regressors.

The system of equations in (2.6) is overdetermined. A typical time series has between 20 and 200 time steps. For each voxel, (2.6) generates one equation for each time step. However, there are only s unknowns stored in $\vec{y}$. The value of s is usually between 1 and 10. Hence, an exact solution cannot be achieved in general.

Instead, a convenient and meaningful decomposition can be found by solving (2.6) in the least-squares sense. That is, the GLM assigns values to $[y_1, y_2, \ldots, y_s]$ to minimize the sum of squared residuals,

$$\min_{\vec{y}} \sum_i \left(\vec{f}_i - [\vec{y}\,\mathbf{B}]_i \right)^2 \quad \Leftrightarrow \quad \min_{\vec{y}} \|\vec{f} - \vec{y}\,\mathbf{B}\|^2 \ , \tag{2.7}$$

where $\|\cdot\|$ represents the Frobenius norm[1], and the notation $[\vec{y}\,\mathbf{B}]_i$ represents sample i of the time series $\vec{y}\mathbf{B}$. From equation (2.6), we can derive the solution to the least-squares problem in (2.7) analytically:

$$\vec{f} \approx \vec{y}\,\mathbf{B} \tag{2.8}$$

$$\vec{f}\,\mathbf{B}' \approx \vec{y}\,\mathbf{B}\mathbf{B}' \tag{2.9}$$

$$\vec{y} \approx \vec{f}\,\mathbf{B}' \left(\mathbf{B}\mathbf{B}'\right)^{-1} \ . \tag{2.10}$$

The matrix $\mathbf{B}'\left(\mathbf{B}\mathbf{B}'\right)^{-1}$ is called the *matrix pseudo-inverse* of $\mathbf{B}$, and will be denoted $\mathbf{B}^\dagger$. Equation (2.10) states that as long as $\mathbf{B}\mathbf{B}'$ is invertible, the least-squares solution can be calculated simply by matrix inversion and multiplication. The matrix $\mathbf{B}^\dagger$ is not invertible when the regressors are not linearly independent [92]. For example, when one regressor is the sum of two other regressors, the decomposition is no longer well-defined, and $\mathbf{B}\mathbf{B}'$ is singular.

The activation detection GLM is solved for each voxel independently. However, $\mathbf{B}$ does not change from voxel to voxel, so the activation detection problem of all voxels can be formulated into a single matrix expression. Recall the definition of a dataset matrix, outlined in section 1.1. A dataset with n volumes can be stored in a single dataset matrix with n columns, each column holding one of the volumes. Each row of the matrix, then, holds the time series for one of the m voxels in the dataset. Since $\mathbf{F}$ has m rows representing m voxels, the matrix of linear fit coefficients also must have m rows, one set of coefficients for each voxel. Hence, we define $\mathbf{Y}$ as just such a matrix, having dimensions $m \times s$. Using this notation, the activation detection problem can be stated for the entire dataset as

$$\mathbf{F} \approx \mathbf{Y}\mathbf{B} \quad \Leftrightarrow \quad \min_{\mathbf{Y}} \|\mathbf{F} - \mathbf{Y}\mathbf{B}\| \ . \tag{2.11}$$

Notice that for notational simplicity we have dropped the exponent from the norm (compare (2.11) to (2.7)) since the minimum of the Frobenius norm and the minimum of the

[1]The Frobenius norm of a matrix is defined as the square root of the sum of the squares of its elements.

square of the Frobenius norm correspond to the same solution. Equation 2.11 has the solution

$$\mathbf{Y} = \mathbf{F}\mathbf{B}^{\dagger} \ , \tag{2.12}$$

where we use an equal sign instead of the approximation symbol "≈" because we are assigning a value to $\mathbf{Y}$.

It is sometimes desirable to specify a particular dataset baseline component. That is, rather than having the GLM determine an optimal linear fit coefficient for the constant regressor, one may wish to override this option and manually specify the baseline component explicitly. The other components are then fitted to the remaining difference. By removing the constant regressor from $\mathbf{B}$, we can replace the baseline with a chosen volume. Define $\mathbf{G}$ as a matrix that holds a copy of the baseline volume in each of its n columns. The corresponding least-squares activation detection problem is then

$$\mathbf{F} \approx \mathbf{G} + \mathbf{Y}\mathbf{B} \quad \Leftrightarrow \quad \min_{\mathbf{Y}} \|\mathbf{G} + \mathbf{Y}\mathbf{B} - \mathbf{F}\| \ , \tag{2.13}$$

yielding the solution

$$\mathbf{Y} = (\mathbf{F} - \mathbf{G})\mathbf{B}^{\dagger} \ . \tag{2.14}$$

Fourier Methods

The Fourier transform is a linear operator that converts image data into an equivalent frequency-domain representation (see appendix B for a discussion of the properties of the Fourier transform). Since the Fourier transform is a linear operator, solution of the GLM can also be carried out in the frequency domain [53]. For each time series, one can take the Fourier transform to obtain its spectral characteristics. In particular, a graph of the magnitude of the Fourier transform shows peaks that correspond to the frequency of repetition of dominant features in the temporal data. For example, a cardiac signal from an electrocardiogram (EKG) repeats a similar pattern. If such data was sampled for a number of heartbeats, the Fourier transform of the data would show a peak at the frequency that corresponds to the person's heart rate. Thus, if the stimulus is presented in a repeating fashion with frequency λ, then a voxel exhibiting behaviour synchronized with that stimulus should show a peak in its Fourier transform at frequency λ. The classification of voxels in the activation map can be made based on the Fourier coefficient corresponding to the base frequency of the stimulus.

An advantage to this method is that it largely decouples the influence of other factors from that of the stimulus, as long as their frequencies do not overlap. An important example of this frequency constraint comes from the effects of respiration and cardiac pulsation. It is well-known that breathing causes motion and also affects the static magnetic field [48]. The cardiac cycle causes motion as well as changes in the blood flow rate. All these factors affect the signal acquired in an EPI sequence. However, since we know the general frequencies of both of these processes, we can decouple the fMRI analysis from their influence. One of the most important factors to consider in the design of an fMRI experiment is the frequency of stimulus presentation. If, for example, the stimulus cycle lasts for 6 seconds, then the neurologically relevant peak in the signal's Fourier transform may coincide with the peak frequency of the respiratory signal (since 6 seconds is a typical period for the breathing cycle). This overlap makes it very difficult to distinguish respiratory effects from activation. As long as the frequency of the stimulus is sufficiently different from the frequency of the physiological effects, this overlap does not pose a problem.

A second advantage to Fourier analysis is that there is no particular assumption made about the shape of the hemodynamic response, except that it repeats at the frequency of the stimulus. The hemodynamic response may be different for different subjects, as well as different for distinct regions of the brain. A particular nuisance is the delay of the BOLD response, the time elapsed after the start of the stimulus until the signal begins. Most other GLM methods have to make explicit assumptions about the delay. However, this delay does not affect the magnitude of the Fourier coefficients.

Another interesting feature of this type of analysis is the opportunity to perform multiplexed experiments at different frequencies. Bandettini *et al.* [8] demonstrated this ability using a finger-tapping paradigm. A subject was prompted to tap the fingers of their right hand for 6.25 seconds, followed by 6.25 seconds of no tapping. Simultaneously, the subject was prompted to do the same thing with their left hand, but at 10-second intervals. In the resulting analysis, the Fourier transform of the time series for voxels in the right motor cortex showed a peak at 0.08 Hz (period of 12.5 seconds). Voxels corresponding to the left motor cortex showed a peak at 0.05 Hz (period of 20 seconds).

One disadvantage to using this method is that it limits the design of the experiment, requiring that the stimulus be applied in a periodic manner.

Deconvolution

Another special kind of general linear model is *deconvolution* [92]. Deconvolution refers to the inverse operation of convolution. As stated earlier, the expected hemodynamic response for a given experiment may be approximated by a convolution of the stimulus function with the BOLD signal (see Fig. 1.3). While the stimulus function is known, the exact shape of the BOLD signal curve is not. Deconvolution attempts to derive the BOLD signal for each voxel. That is, it estimates the BOLD signal so that its convolution with the stimulus function yields a time series that closely models the observed voxel time series.

This BOLD curve can be calculated by constructing a GLM that includes a number of delayed stimulus functions as regressors. Thus, each voxel time series is modeled as a linear combination of these delayed functions, the weights of which are used to derive the BOLD curve.

2.2.4 Real-Time fMRI

There are many stages of numerical processing that have to be performed to attain an activation map from an fMRI experiment. Many of those methods were discussed in section 2.2.2. An fMRI experiment that is contaminated with such artifacts is of little or no value, yielding activation maps that are obviously related to the artifacts, not to the neurological activity.

Until recently, all fMRI data processing was done off-line, long after the subject had left the scanner. One advantage to being able to complete the processing and observe the activation maps quickly is that a failure can be detected while the subject is still in the scanner, allowing the option to re-do the experiment immediately. This option is of great use, given the shortage of scanning resources.

Furthermore, fMRI studies often consist of several experiments performed in sequence. The intermediate fMRI activation results give the investigator feedback that can be used to direct subsequent experiments. For example, a set of experiments could be designed using a tree-structure paradigm, in which results from a given experiment can determine which branch to follow.

Until recently, "real-time" fMRI processing involved reconstruction, and motion correction, as the data was being collected. Since the full time series was needed for the activation detection step, this process was spawned once the experiment was completed [28, 57, 90]. The results were then presented within a few minutes.

One of the most significant breakthroughs in real-time fMRI processing was the discovery of a way to incrementally update the statistical correlation as each new volume was acquired [22]. The technique sidesteps the need to have the entire sequence of volumes before starting the correlation analysis. Instead, the method can add the newly arrived volume to the current statistical parametric map, requiring very little processing. Since the discovery of this method, other methods have been proposed that incrementally update a sliding-window statistical correlation operation [37, 66]. The correlation uses the last N volumes (typically 20 volumes), casting off the oldest volume every time a new one is received.

2.2.5 Statistical Inference

It is not uncommon to see a voxel time series that is highly correlated with the stimulus even when the voxel is clearly not activated. For example, voxels in the scalp, or even outside the head, can be correlated simply by chance. This phenomenon is simply the result of performing activation detection on thousands of voxels. It is most likely that a few of them will be correlated just by random chance alone. For this reason, so called "correlated" voxels often undergo an additional battery of statistical tests to try to establish whether their time-series variation truly is caused by the stimulus, or if the variation can be explained by random chance.

One such test is the Students T test. It generates a likelihood that the observed time series is explainable by the null hypothesis. The null hypothesis states that the time series contains no influence from the stimulus. This statement is translated into a mathematical expression as follows. Recall that if $\vec{f}$ is the voxel time series, $\mathbf{B}$ is a matrix of regressors, and $\vec{y} = [y_1, y_2, \ \ldots \ , y_s]$ is a row-vector of associated coefficients, then the general linear model is

$$\vec{f} \approx y_1\vec{b}_1 + y_2\vec{b}_2 + \cdots + y_s\vec{b}_s \tag{2.15}$$

$$\vec{f} \approx \vec{y}\,\mathbf{B}\ . \tag{2.16}$$

For the sake of argument, assume that $\vec{b}_1$ and $\vec{b}_2$ correspond to two stimulus regressors, each representing a different condition. The remaining regressors represent corrections such as time-course detrending. If the null hypothesis states that the voxel reacts the same to condition 1 as to condition 2, then we would expect $y_1 = y_2$. The contrast between

conditions 1 and 2 can be evaluated by letting $\vec{c} = [-1, 1, 0, \ldots]$, and then assigning a value to t using,

$$t = \frac{\vec{c} \cdot \vec{y}}{\sigma^2} \ , \tag{2.17}$$

where σ^2 is the standard error for the voxel time series, and can be computed easily [34]. The quantity t has the Students T distribution. Roughly speaking, the t-value represents the same thing as a z-score for a Gaussian distribution. That is, if the hypothesis is true, then we would expect to see values of t near zero most of the time. Values of t that are far from zero are very unlikely to occur if the hypothesis is true. Thus, observation of a large (positive or negative) t-value is evidence that the null hypothesis is *not* true. Voxels for which such evidence is observed are labeled as "active" in the contrast between the conditions in question. A t-value can be calculated for every voxel, and the activation map is thus a simple thresholding, rejecting all voxels with t-values close to zero. Typically, if a voxel's t-value is outside the 95% probability interval, then that voxel's time series has a probability of occurring of less than 5% if the null hypothesis is true. Thus, for that voxel, we reject the null hypothesis, and instead consider the voxel to be active.

The above statistical analysis is univariate; it assumes every time series is independent of all others. However, adjacent voxels are heavily correlated, and are thus dependent. Multivariate statistical analysis factors in this spatial information, and derives the likelihood that a voxel (and its neighbourhood) are truly activated by the stimulus [34, 53]. Description of these methods is beyond the scope of this thesis.

2.3 Motion Correction

Given that motion is likely to occur during an imaging sequence, it is appropriate that much research has been focused on how to manipulate the data to compensate for the motion, and possibly reverse its effect. Many such methods are similar to those used in correcting for physiological artifacts. However, the motion correction techniques discussed here are designed to address artifacts caused by gross subject motion, not physiologically-based motion.

The general motion-correction, or registration problem can be posed as follows. Given two volumes, $\mathbf{f}$ and $\mathbf{g}$, we wish to find the motion parameters $\mathbf{x}$ that define a p-degree-of-freedom transformation, $T^{-1}(\mathbf{f}, \mathbf{x})$, such that $T^{-1}(\mathbf{f}, \mathbf{x}) \approx \mathbf{g}$. For 3D rigid-body motion, p is usually six: three translations and three rotations. This problem is highly over-determined,

so the solution consists of minimizing some measure of the difference between $T^{-1}(\mathbf{f}, \mathbf{x})$ and $\mathbf{g}$,

$$\min_{\mathbf{x}} C\left(\mathbf{g}, T^{-1}(\mathbf{f}, \mathbf{x})\right) \ , \tag{2.18}$$

in which the cost function C can take on many forms, and will be discussed in section 2.3.3. An equivalent formulation is to apply the transformation to $\mathbf{g}$ rather than $\mathbf{f}$, which yields the expression

$$\min_{\mathbf{x}} C\left(T(\mathbf{g}, \mathbf{x}), \mathbf{f}\right) \ . \tag{2.19}$$

2.3.1 Gross Subject Motion

There are many different kinds of motion that can occur during an imaging sequence. Tissue can shift, rotate, stretch and deform. However, for head MR imaging sequences, the motion is often assumed to be rigid, involving only translations and rotations (a correction technique for scaling is also presented in [107]). This type of motion is called *rigid-body motion*.

Motion can affect the images in different ways, depending on when the motion takes place with respect to the scanning sequence. Also, different types of scan sequences are susceptible to different types of motion artifacts because the scanning times vary widely. However, most types of motion can be placed into one of two categories: *inter-scan* and *intra-scan* motion.

Intra-scan motion is a type of motion artifact that is commonly observed in scanning sequences that take a long time to acquire. This motion happens during the acquisition of a single k-space image. Since the data is collected in k-space, motion during the sampling of a slice causes the k-space data to be inconsistent. After the inverse Fourier transform, the image does not accurately reflect the volume in the scanner, and often contains artifacts such as blurring and edge ringing.

Inter-scan motion is any motion that is not intra-scan motion. Since intra-scan motion occurs during the acquisition of a k-space slice, inter-scan motion occurs between the acquisition of k-space slices. Moreover, in fMRI where volumes are acquired quickly, motion is often assumed to occur only between volumes, causing the volumes in the time series to be mis-aligned.

A comprehensive study of the effects of motion on fMRI activation maps was conducted by Hajnal *et al.* [41]. In their study, a single slice was repeatedly acquired using a gradient-echo sequence. As the sequence of slices was being acquired, the patient was presented with

visual and motor stimuli. Two-dimensional motion detection was executed to estimate the subject's motion. These motion estimates were then used to create a second, simulated sequence of images with the same motion by duplicating and resampling a single baseline image once for each image in the original sequence. The simulated image sequence, then, had the same motion, but lacked any other fluctuations such as BOLD contrast. In seven out of eight experiments performed, significant stimulus-correlated motion was detected. The activation maps produced from the simulated data were very similar to those produced from the actual data. This study illustrates the overwhelming impact that motion can have on the discovery of activated voxels. It also shows that motion cannot be assumed to be independent of the stimulation paradigm.

In a study by Ramsey *et al.* [83], fMRI activation maps were shown to be more accurate after 3D motion correction. In their study, a sequence of 3D multi-echo volumes was acquired with navigator echoes (see section 2.3.2 for a description of navigator echoes). A variety of motion correction techniques were tested, and it was found that these corrections reduced the signal variation by 20% and the extent of activation was more stable between multiple scans of a subject. Similar results have since been demonstrated in [66, 70].

Depending on the timing of motion, a number of different artifacts can result. It is common to acquire k-space one line at a time. Each line is acquired quickly, and is usually assumed to be motion-free. However, motion can occur between k-space lines, causing inconsistencies in the k-space data. The resulting motion artifacts are called *intra-scan* motion artifacts. Subject translation affects the phase of the samples, while subject rotation causes the corresponding samples in k-space to be rotated (see appendices B and A for properties of the Fourier transform, and how the Fourier transform is linked to k-space acquisition). After making the appropriate corrections to the k-space data, some regions can be over-sampled, while other regions can be under-sampled. Figure 2.4 depicts k-space after being corrected for two sudden patient movements. This differential sampling can pose a problem to the regridding and reconstruction process, since the Nyquist criterion is not necessarily satisfied everywhere (for an overview of the effects of non-uniform k-space sampling issues, see [88]). Research has been done to overcome this problem by using oversampling and replacing regridding with specialized, more stable matrix inversion. Super-sampling by a factor of four was shown to be sufficient to remove most phase errors from diffusion-weighted images [7].

Figure 2.5 shows the intra-scan motion artifact for the k-space shown in Fig. 2.4. Notice

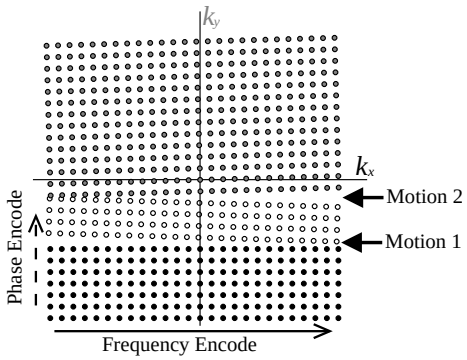


Figure 2.4: Motion corrected k-space. In this example, the subject moved at two different times during the scan, dividing the k-space lines into three blocks. The samples have been phase modulated and rotated to correct for the motion. The change in the shade of the samples indicates phase modulation. Some regions are over-sampled, while others are under-sampled.

the blurry smudge-like areas shown in the motion-corrupted image.

Motion that occurs between completed volumes is called inter-scan motion. Inter-scan motion can confound the analysis of a time series of images and is a major concern in fMRI analysis. If the volumes are mis-aligned, the time series for adjacent voxels can be mixed. For example, if an active voxel is beside an inactive voxel (that is, a voxel with activity not correlated with the stimulus), head translation of a single voxel half-way through the experiment can replace the second half of the time series for the active voxel with the time series of the inactive voxel. The statistical analysis will be inconclusive for these voxels. Furthermore, motion can cause significant false-positive identification of voxels, particularly if the motion is correlated to the stimulus. For example, if the subject is asked to press a button with the presentation of a certain stimulus, slight head motion can be caused by the hand motion. This motion, which is clearly correlated to the button-press stimulus, can make voxels near high-contrast borders appear to be active. When the head relaxes, the voxel may be in a dark region. But when the head moves due to the button press, the voxel may fall in a near-by bright region. There may be no activation, but the stimulus-induced motion of the voxel from a dark area to a bright area creates the impression of activation.

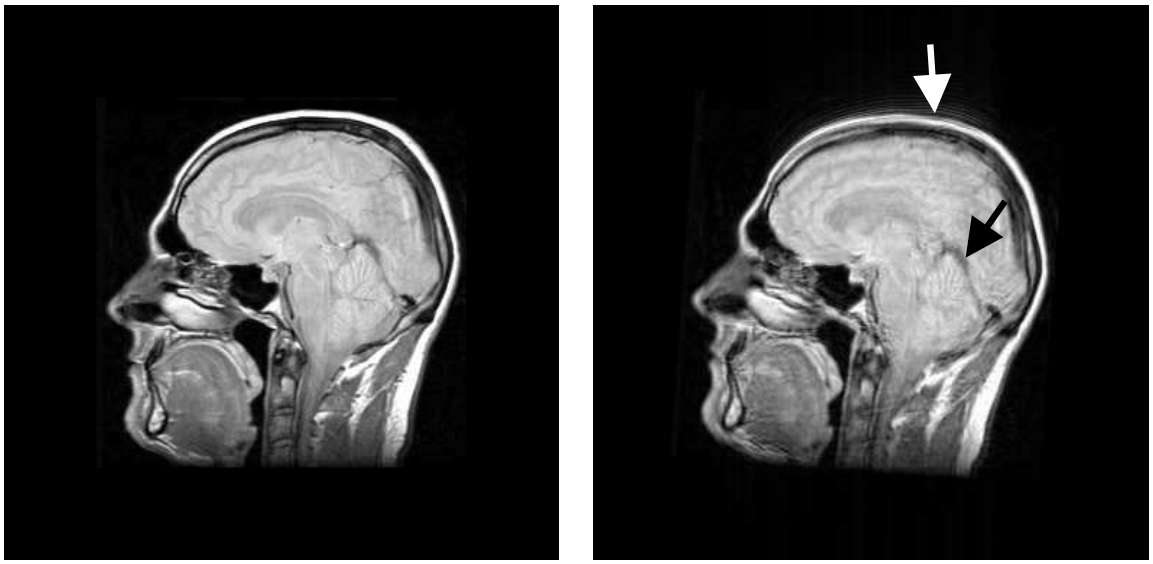


Figure 2.5: An artifact-free image is on the left, while the same image with intra-scan motion artifacts is shown on the right. The motion artifact was created by adding two instantaneous movements to the k-space data. One movement was a translation of (3,4) pixels and rotation of 6°, and the other was a translation of (-5,-6) and a rotation of $-10°$. Notice the "ringing" at the high-contrast region at the top of the head, and the general blurring of other structures.

2.3.2 Navigator Echoes and Prospective Motion Correction

Navigator echoes, or NAV's, are small sets of k-space samples that are acquired between imaging excitations. Their purpose is to gain an estimate of the object's position. The echoes commonly consist of a few lines in k-space. These ultra-fast acquisitions are interleaved throughout the scanning sequence.

Navigators are useful in intra-scan motion correction in MRI because they offer a repeated set of measurements that can be used to estimate motion [24, 36, 51, 69, 77, 83, 84, 91]. In many cases, the NAV consists of a 1D "pencil-beam" acquisition. A slice of tissue is excited by an RF pulse, after which a single gradient is used to induce phase encoding along one direction of the slice. In an imaging sequence, this first phase encoding gradient would be followed by a second "frequency-encoding" gradient, facilitating resolution of the signal to locations in the 2D slice. However, pencil-beam NAV's do not utilize a frequency-encoding gradient. As a result, the signal captured by the NAV corresponds to a 1D segment along one of the axes in the frequency domain (depending on the slice-select and phase-encoding gradients). For example, suppose a y-gradient was used to excite a slice in the coronal plane. Then, suppose a z-gradient was used to apply phase-encoding in the inferior-superior direction. The resulting MR signal corresponds to sampling k-space along the k_z-axis. The Fourier Projection theorem states that the inverse Fourier transform of this 1D signal is the Radon projection of the slice onto the z-axis. In this example, the projection can be used to observe the inferior-superior motion of tissue [91]. Ehman and Felmlee first demonstrated this concept in 1989 [24] by moving a gel-filled phantom in the MR scanner, and collected navigator echoes while imaging. They then used the motion information from the NAV's to correct the phase of the k-space views, resulting in a substantial decrease in motion blur.

The motion information captured by navigator echoes can be displayed elegantly by stacking the NAV's in a hybrid space creating an image with the spatial dimension along one axis, and time along the other. Figure 2.6 shows a hybrid-space image for a scanned phantom [36]. The horizontal axis corresponds to the z-axis in the scanner, while the vertical axis corresponds to time. Motion of the phantom in the inferior/superior direction is obvious by the horizontal displacement of image sections.

Navigator echoes are also used for inter-scan registration, usually applied to fMRI scans.

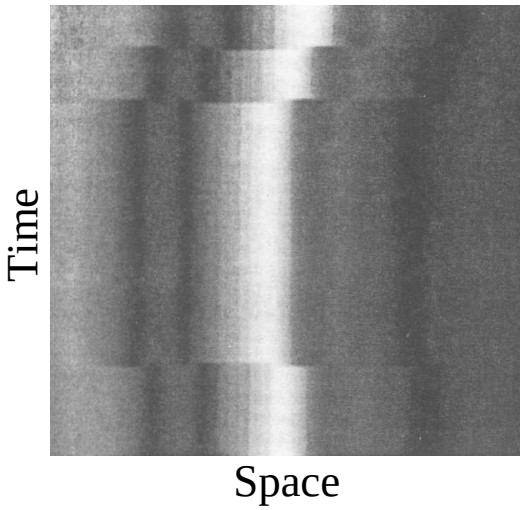


Figure 2.6: Navigator echo hybrid space for a phantom. The horizontal axis is the z-axis, while the vertical axis is time. The NAV's are plotted in chronological order from top to bottom. Three sudden movements are clearly evident. (Taken with permission from Fu *et al.* [36])

Recall that in fMRI, multiple datasets that are almost identical are collected in rapid succession. A navigator echo can be collected between volume acquisitions.

Motion correction that is performed as a post-processing step after the acquisition is complete is called *retrospective* motion correction. This strategy works well as long as there is no inherent loss of information caused by the motion. For example, if only in-plane motion occurs, then essentially all the information for a given slice is maintained in the k-space data for that slice.

In contrast, through-plane motion is a source of imaging artifact that cannot be fully corrected retrospectively. In fast fMRI scanning sequences such as ss-EPI, since the dipoles never reach a fully relaxed state, a voxel's spin history is a factor in its resulting signal. In ss-EPI, the next RF pulse is applied before the net magnetization vector's z-component has relaxed to its full length. Thus, the resulting M_{xy} is proportionately smaller. After a few excitations, the cycle of excitation and relaxation converges to periodic behaviour, exhibiting the same M_z before each excitation. However, if this pattern is disrupted, the signal may not reflect the appropriate steady-state dynamics. For example, suppose a volume composed of 35 sequentially-acquired slices is captured every 3.5 seconds (100 ms per slice). Then, each

voxel is excited once every 3.5 seconds. If a voxel experiences through-plane motion, shifting from one plane to its adjacent plane, then the voxel's next TR will be either 3.4 seconds, or 3.6 seconds, depending on which slice the voxel moved to. The change in the TR affects the length of M_z, and results in a fundamental loss of information in the k-space data. For this reason, even perfect re-alignment cannot compensate for spin history errors caused by through-plane motion.

Friston *et al.* [35] proposed a post-processing first-order correction method to reduce these errors. The method involves first estimating the patient motion, and then decomposing each voxel time series into two orthogonal components: a motion-artifact-free component, and a motion-related component. The decomposition depends on the motion estimates of not only the current volume, but also on those of the previous volume. Functional MRI analysis based on the motion-artifact-free component showed a substantial improvement in the sensitivity of activation maps.

An alternative to dealing with spin history disruption is to alter the scan parameters in real-time to move the imaging plane to match the motion of the subject. This type of motion correction is called *prospective* motion correction.

Lee *et al.* [56] used a pencil-beam NAV to monitor inferior-superior translation of the head during an fMRI scan. A specialized high-speed computer performed the processing of the NAV and updated the scan parameters so that the imaging plane was shifted with the subject before the next RF pulse. By doing so, the slice-selection plane was properly placed even though the subject moved. However, their technique addressed only inferior-superior translational motion. In a different study, Lee *et al.* [55] proposed the use of *orbital navigator echoes*, or ONAV's, to determine rotational motion as well. An OVAV is a 1D navigator echo that follows a circular trajectory in k-space, usually in one of the principal scanner planes. Magnitude information in k-space can be used to derive object rotation, while phase information can be used to derive translations (see appendix B). Hence, the magnitude and phase of an ONAV can be compared to that of a reference ONAV to determine the amount of rotation and translation. Two ONAV's, one in each plane perpendicular to the imaging plane, can sense two rotations and all three translations of rigid-body motion. However, due to hardware limitations, only through-plane motion was corrected prospectively [55] while in-plane translation and rotation was corrected retrospectively. The corrected data again showed improved fMRI sensitivity.

A further improvement to the prospective ONAV strategy was published by Ward *et*

al. [93], allowing for prospective correction of all motion except in-plane translation. Three ONAV's were used to achieve this result. Ward also determined that choosing a navigator slice thickness that encapsulated the entire object kept the object's tissue excitation history homogeneous, preventing multiple NAV's acquired in rapid succession from interfering with each other's spin histories. Ghost reduction techniques were subsequently demonstrated on the prospectively corrected EPI data [94].

The natural extension of orbital navigator echoes is to acquire *spherical navigator echoes*, or SNAV's. Like ONAV's, SNAV's use magnitude information to derive rotations, and phase information to derive translations. The feasibility of SNAV's was established by Welch *et al.* [96, 97], in which a navigator echo traversed the surface of a sphere in k-space. Comparing two SNAV's requires the solution of two 3-degree-of-freedom optimization problems: one to solve for the three rotations, and one to solve for the three translations. Although the technique was demonstrated to be reasonably accurate, a prospective implementation has not yet been developed.

While the use of navigator echoes is very promising, they have not yet been integrated into fMRI scanning sequences. Part of the reason for this reluctance is that EPI is a fast scanning sequence, and repeatedly acquires images of the same tissues. Contrast this situation to acquiring a T_1-weighted anatomical slice where the same information is never acquired more than once. The redundancy built into fMRI scanning sequences fulfills part of the role of NAV's. Another reason navigator echoes are not used in fMRI is that the EPI scanning sequence attains a non-equilibrium steady state. The excitation cycle is so fast that the dipole spins in the tissue never fully relax to their equilibrium state. Instead, the cycle of partial relaxation and excitation settles into a steady state. Adding an RF pulse for a NAV would disrupt this steady state.

For fMRI, however, Thesen *et al.* [87] proposed a prospective motion compensation method that does not use navigator echoes. They demonstrated that spin history artifact reduction can be achieved in fMRI by prospectively adjusting the imaging plane with a one-volume delay. Their strategy was to perform motion detection on the most recently-acquired volume, and adjust the scan plane accordingly. Naturally, motion can occur in the delay between the end of one volume acquisition, and the beginning of the next. However, Thesen notes that if the subject's motion is reasonably smooth and slow, the motion between adjacent volumes will be small. Correcting the imaging plane with a motion estimate that contains only a small error greatly reduces the effect of through-plane spin history artifacts.

However, registration still needs to be done on the dataset to correct for the remaining one-volume motion mis-alignment. This final registration set is carried out retrospectively.

2.3.3 Cost Functions

When performing image registration, it is necessary to have a way to quantify the similarity between two volumes. That is, given two volumes, how does one numerically represent the degree to which they match? The function that determines this quantity is called a *cost function*.

Some cost functions used in medical image registration use fiducials in their calculation [25, 51, 67]. Others use various anatomical features such as vasculature [68], sulci and gyri [17, 89], or tissue surfaces [58]. However, here we consider only volume intensity-based cost functions: those which compare the intensities of the volumes on a voxel-by-voxel basis. In a study by West *et al.* [98], volume-based inter-modality registration methods were shown to be more accurate than surface-based registration methods. Furthermore, our focus will be on rigid-body motion.

Five common intensity-based cost functions used in registration today are correlation, sum of squares, entropy, robust estimators, and ratio image uniformity. These cost functions are described below.

Correlation

One of the first cost functions used in medical image registration was *correlation* [103]. The Pearson product-moment cross correlation coefficient, CC, is defined as

$$CC(\mathbf{f}, \mathbf{g}) = \frac{\sum_i \mathbf{f}_i \mathbf{g}_i}{\left(\sum \mathbf{f}_i^2 \sum_i \mathbf{g}_i^2\right)^{\frac{1}{2}}} \ , \tag{2.20}$$

where $\mathbf{f}_i$ represents the intensity of voxel i in volume $\mathbf{f}$. Variations of the definition include multiplication by a scalar such as $\frac{1}{N}$ for N voxels. It is also common for the images to first be neutralized by subtracting their average intensity, making the mean voxel intensity of each image zero.

If two volumes, $\mathbf{f}$ and $\mathbf{g}$, are identical, then the sum in the numerator is the same as each of the sums in the denominator, and the value of CC is 1. This maximum possible value indicates that the two functions (volumes) are perfectly correlated; when one goes up (down), the other goes up (down). Notice, however, that this maximum value can be

achieved not only when the volumes are identical, but also when their intensities differ by a global scaling factor [44]. Let α be a non-zero scalar value. Then

$$\begin{aligned} CC(\mathbf{f}, \alpha\mathbf{f}) &= \frac{\sum_i \mathbf{f}_i \alpha \mathbf{f}_i}{\left(\sum \mathbf{f}_i^2 \sum_i (\alpha \mathbf{f}_i)^2\right)^{\frac{1}{2}}} && (2.21) \\ &= \frac{\alpha \sum_i \mathbf{f}_i \mathbf{f}_i}{\left(\alpha^2 \sum \mathbf{f}_i^2 \sum_i \mathbf{f}_i^2\right)^{\frac{1}{2}}} && (2.22) \\ &= \frac{\sum_i \mathbf{f}_i^2}{\left(\sum \mathbf{f}_i^2 \sum_i \mathbf{f}_i^2\right)^{\frac{1}{2}}} && (2.23) \\ &= 1\ . && (2.24) \end{aligned}$$

When aligning two 1D images (functions), a common strategy is to evaluate the correlation coefficient at regularly-spaced values of the motion parameter, selecting the parameter value that achieves the highest CC value [24, 84, 91]. For example, many navigator echoes are 1D, and the task is to determine the optimal translation that will align the floating NAV with the base NAV. The floating NAV is shifted one pixel at a time. For each offset, the CC is calculated, resulting in a graph plotting CC as a function of offset. The offset that corresponds to the highest CC value is the optimal translation.

Similar to the strategy above, Pipe used the correlation cost function to find the optimal 2D rigid-body transform between segments of the PROPELLER MRI method [79]. In this method, k-space is populated by rectangular segments that radiate from the centre of k-space like propellers on a plane. Each segment is acquired during one TR and subject motion is assumed to occur only between segments. The overlapping parts of the segments (near the centre of k-space) are compared to detect motion. Trial rotations are applied to the floating segments. For each rotation, the CC is calculated. A parabola is fit to the resulting angle versus CC curve. The maximum of the parabola is chosen as the optimal re-alignment angle.

To determine the translation, Pipe used a particularly efficient optimization method that is specific to the correlation cost function. The operation of evaluating CC for different offsets (translations) is actually a convolution. To see this in the 1D case, let d be the pixel translation for the floating image, $\mathbf{g}$. Then, CC_d is a function of d, and we have

$$\begin{aligned} CC_d(\mathbf{f}, \mathbf{g}) &= \frac{\sum_i \mathbf{f}_i \mathbf{g}_{i-d}}{\left(\sum \mathbf{f}_i^2 \sum_i \mathbf{g}_{i-d}^2\right)^{\frac{1}{2}}} && (2.25) \\ &= \frac{\sum_i \mathbf{f}_i \mathbf{g}_{i-d}}{\|\mathbf{f}\|\ \|\mathbf{g}\|}\ . && (2.26) \end{aligned}$$

It should be noted that the substitution of $\|\mathbf{g}\|$ for $\left(\sum_i \mathbf{g}_{i-d}^2\right)^{\frac{1}{2}}$ assumes that the translation of d does not cause any of the imaged object to leave the field of view. Notice that if we reverse $\mathbf{g}$ by negating its subscript, the numerator of (2.26) becomes the convolution in d

$$\sum_i \mathbf{f}_i \mathbf{g}_{d-i} \ . \tag{2.27}$$

Convolution in the spatial domain is equivalent to multiplication in the frequency domain. That is, by taking the discrete Fourier transforms of $\mathbf{f}$ and $\mathbf{g}$, the convolution in the correlation formula can be achieved by an efficient one-time multiplication. Let $\hat{\mathbf{f}}$ be the frequency domain representation of $\mathbf{f}$, and $\hat{\mathbf{g}}$ be the frequency domain representation of the reversed $\mathbf{g}$. Then, finding the d that maximizes $CC_d(\mathbf{f}, \mathbf{g})$ is the same as finding the d that maximizes the inverse discrete Fourier transform of

$$E_d(\hat{\mathbf{f}}, \hat{\mathbf{g}}) = \hat{\mathbf{f}}_d \, \hat{\mathbf{g}}_d \ . \tag{2.28}$$

The cost of evaluating $E_d(\hat{\mathbf{f}}, \hat{\mathbf{g}})$ for a given d is very small once $\hat{\mathbf{f}}$ and $\hat{\mathbf{g}}$ are calculated. This same principle works in higher dimensions. Consider the savings when finding the optimal translation in 2D. We want to find the ordered-pair (a, b) that maximizes

$$CC_{(a,b)}(\mathbf{f}, \mathbf{g}) = \frac{\sum_i \sum_j \mathbf{f}_{(i,j)} \mathbf{g}_{(i-a,j-b)}}{\|\mathbf{f}\| \ \|\mathbf{g}\|} \ . \tag{2.29}$$

Suppose $\mathbf{f}$ and $\mathbf{g}$ are 2D images, each having dimensions $N \times N$. The algorithmic complexity of evaluating $CC_{(a,b)}$ for a particular (a, b) pair using (2.29) is N^2. There are N^2 different (a, b) pairs, making the complexity of evaluating $CC_{(a,b)}$ for all possible (a, b) pairs equal to N^4. On the other hand, performing the discrete Fourier transform of each image has complexity $N^2 \log N$, and the multiplication in the frequency domain has complexity N^2. Thus, the algorithmic complexity of evaluating $CC_{(a,b)}$ for all possible (a, b) pairs is $\mathcal{O}(N^2 \log N)$. This can result in a huge decrease in computation. However, if it is known that the optimal displacement is small, then the convolution does not need to be calculated for every (a, b) pair. Instead, the search can be focussed onto a small number of possible displacements. If that number of trial displacements is k, then the complexity of finding the optimal displacement is kN^2. This may be more efficient than performing the convolution in the frequency domain, depending on the number of trial displacements.

This frequency-domain implementation of the correlation cost function has been demonstrated in a number of motion correction publications [26, 61, 79, 80].

Entropy

Entropy is a measure of disorder. It can also be interpreted as a measure of information content. As the amount of entropy (disorder) in a function increases, the amount of information increases.

The formula for entropy is

$$H(\mathbf{f}) = -\sum_{i=1}^{N} p_i \log p_i \ , \tag{2.30}$$

where $\mathbf{f}$ is a discritized function with values ranging from 1 to N, and p_i is the probability that a randomly-chosen element (pixel) has value i ($i \in [1, N]$). Notice that $H(\mathbf{f})$ is never negative. Suppose a gray-level image contains 256 gray levels, ranging from 1 to 256. The sum in (2.30) goes from 1 to 256, while each p_i is the fraction of the total number of pixels in the image that have intensity i.

Suppose the contrast of the image is severely reduced so that it is difficult to distinguish different gray-tones. As the pixel intensities converge toward a constant value, all the p_i's converge to 0 except for one, which converges to 1. Thus, $H(\mathbf{f}) \to 0$ (since $\lim_{p_i \to 0^+} p_i \log p_i = 0$ and $\log 1 = 0$). Moreover, zero is the minimum possible entropy value, since entropy is never negative. This is consistent with the notion that a blank (constant) image has no information. Now suppose that our function contains all of the different intensities, all with equal probability. Then, $H(\mathbf{f}) = -N \frac{1}{N} \log \frac{1}{N} = \log N$. This is the maximum possible entropy value for N different intensities. It corresponds to the maximum disorder, or maximum amount of information.

Minimizing the entropy of an image tends to simplify it, favouring images with regions of nearly-constant intensity, separated by jump-discontinuities. The intensity histogram of the image becomes more tightly-packed, with higher, thinner peaks.

As an example, consider a gray-tone image with very low contrast. In such an image, it is not difficult to predict what gray-level a randomly-chosen pixel will have since all the pixels have similar intensities. Thus, it is predictable and contains little information. On the other hand, consider an image in which the contrast is high. Then, given a randomly-chosen pixel, predicting the gray-level is quite difficult due to the variety of intensities present. This image is less predictable, and hence contains more information. These ideas form the cornerstone of the entropy measure.

Entropy is used as a focus criterion for intra-scan motion correction. Motion between

phase encode lines tends to blur the resulting image, smearing crisp edges into fuzzy gray-tone gradients. The resulting spreading of the image histogram means the image entropy is increased. Motion correction transformations can be applied to individual phase encode lines in k-space such that the entropy of the reconstructed image is minimized [4, 6, 69]. The same technique can be used on the gradient image (magnitude of the spatial gradient vector). Gradient images can be produced for a variety of different trial corrections. The transformation that minimizes the entropy of the gradient image is used for the motion correction [63, 64, 69].

It may seem counter-intuitive to minimize the information, or entropy, in an image. While the true, uncorrupted image contains an amount of inherent information, our goal is to remove motion artifacts such as duplicated structures, ghosts and smudging, that tend to increase the entropy further. Hence, we favour images in which the intensity histogram contains highly compact peaks. As the histogram becomes more dispersed, the entropy increases, and the image quality is compromised.

In inter-scan registration, entropy can also be applied to the joint probability histogram to evaluate the degree of similarity between two images. Consider two images, $\mathbf{f}$ and $\mathbf{g}$. Their joint histogram, or joint probability distribution, is a matrix in which the $(i,j)^{\text{th}}$ element, $p_{i,j}$, represents the probability that a pixel with intensity i in image $\mathbf{f}$ corresponds to a pixel of intensity j in image $\mathbf{g}$. The entropy of the joint histogram is

$$H(\mathbf{f},\mathbf{g}) = -\sum_{i,j} p_{i,j} \log p_{i,j} \ . \tag{2.31}$$

To gain a better understanding, consider the joint histogram between $\mathbf{f}$ and itself. The joint histogram matrix will contain all zeros except for one's on its diagonal. That is, an intensity of 1 always maps to an intensity of 1 in the other image, and never to 2, or 3, or 4, etc. Likewise, intensity i always maps to i, and never to anything else. Thus, $p_{i,i} = 1$, and $p_{i,j} = 0$ for $i \neq j$.

However, if the second image, $\mathbf{g}$, is a slightly shifted copy of $\mathbf{f}$, then each pixel is shifted to a position in its neighbourhood, and we no longer have the pixel-to-pixel correspondence of intensities. Since many images tend to have regions of slowly-varying gray levels, a pixel of intensity i is often surrounded by pixels with intensities close to i. Hence, while $p_{i,i}$ is likely no longer equal to 1, the values of $p_{i,i-1}$ and $p_{i,i+1}$ are likely no longer equal to zero. This causes a smudging of the histogram, blurring the well-defined crisp diagonal, as seen in

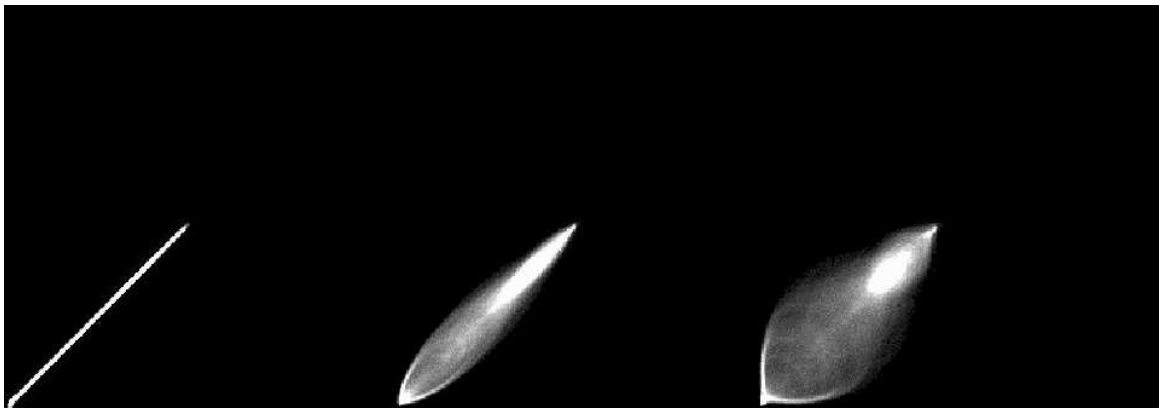


Figure 2.7: The joint histogram of an MRI volume to itself for three different displacements. Going from left to right, the displacements go from small to large. When the volumes are aligned, the histogram is tightly clustered. When the volumes are mis-aligned, the joint histogram spreads out. (Taken with permission from Hill *et al.* [45])

the sequence in Fig. 2.7. Thus, the entropy of the joint histogram is increased by applying motion.

The images being compared need not map intensity i to a nearby intensity. A person can easily recognize the similarity between a black and white photograph, $\mathbf{f}$, and its photo-negative, $\mathbf{f}^*$. The shapes in the images are the same because the mapping of colours from one image to the other is consistent. Black always maps to white and white always maps to black, while the intermediate tones follow the same flip-flop mapping. The joint histogram of this combination of images will also be a diagonal line (perpendicular to that comparing $\mathbf{f}$ to $\mathbf{f}$). Again, a small shift in one of the images causes the joint histogram to smudge, distributing the probabilities off the diagonal, likely to nearby elements.

Finally, the actual *form* of the intensity mapping is not necessarily important when evaluating the similarity between two images. The *consistency* of the mapping is of greater interest. For example, the mapping of intensities between a T_1-weighted MR image and a perfectly registered T_2-weighted MR image is complicated, and most likely non-linear. However, we might expect that if two small volumes of tissue have the same T_1 signal, then they will also have the same T_2 signal, and vice versa. So, any pixel that has a certain intensity i in the T_1 image should have the corresponding intensity j in the T_2 image. This is not technically true, but is a useful approximation for registration purposes. For this reason, joint histogram entropy-based cost functions are often used to align images acquired

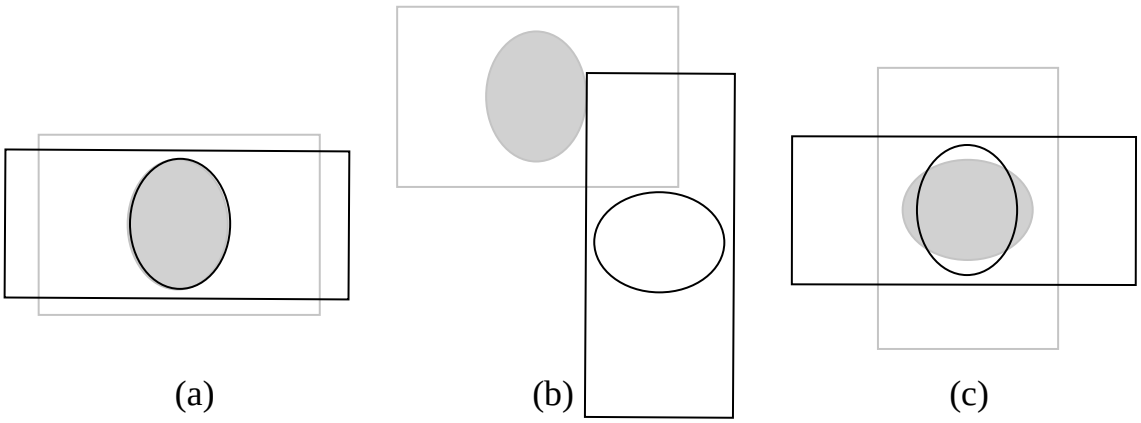


Figure 2.8: Alignment (a) is the correct alignment. However, the entropy of the joint histogram is smaller for (b) because it is calculated only on the overlapped portion. Alignment (c) shows how maximizing mutual information can be inappropriate, since including less background in the overlap increases mutual information. (Adapted with permission from Studholme *et al.* [86])

from different modalities, such as aligning a PET study to an MRI study [100].

One problem with the joint entropy cost function is that the minimum value does not necessarily correspond to the best alignment. Consider the case in which the overlap between the two images includes only a corner of the image, containing only background. This situation is demonstrated in Fig. 2.8(b). Since the entropy measure is applied only to the region of overlap between the two images, the entropy value is very low. The image backgrounds contain little information. Clearly, this is not satisfactory.

This problem led researchers to consider the use of mutual information to measure image similarity. Recall that the term $H(\mathbf{f}, \mathbf{g})$ is the amount of information that is in either $\mathbf{f}$ or $\mathbf{g}$ or both. If you subtract $H(\mathbf{f}, \mathbf{g})$ from the sum of $H(\mathbf{f})$ and $H(\mathbf{g})$, you are left with the total amount of information that both $\mathbf{f}$ and $\mathbf{g}$ share. This is called the mutual information of $\mathbf{f}$ and $\mathbf{g}$. The difference between joint entropy and mutual information can be illustrated using Venn diagrams, as shown in Fig. 2.9. In particular, mutual information is defined as

$$I(\mathbf{f}, \mathbf{g}) = H(\mathbf{f}) + H(\mathbf{g}) - H(\mathbf{f}, \mathbf{g}) \ . \tag{2.32}$$

Note that the marginal probabilities for each image can be derived from the joint probabilities using the relations

$$p_i = \sum_j p_{i,j} \quad \text{and} \quad p_j = \sum_i p_{i,j} \ . \tag{2.33}$$

In other words, the histogram for image $\mathbf{f}$ can be calculated by simply adding down the columns of the joint histogram. Likewise, the histogram for image $\mathbf{g}$ results from adding along the rows of the joint histogram. Thus, the formula for mutual information can be simplified.

$$\begin{aligned}
I(\mathbf{f},\mathbf{g}) &= H(\mathbf{f}) + H(\mathbf{g}) - H(\mathbf{f},\mathbf{g}) && (2.34)\\
&= -\sum_i p_i \ln p_i - \sum_j p_j \ln p_j + \sum_i \sum_j p_{i,j} \ln p_{i,j} && (2.35)\\
&= -\sum_i (\textstyle\sum_j p_{i,j}) \ln p_i - \sum_j (\sum_i p_{i,j}) \ln p_j + \sum_i \sum_j p_{i,j} \ln p_{i,j} && (2.36)\\
&= -\sum_i \sum_j p_{i,j} \ln p_i - \sum_i \sum_j p_{i,j} \ln p_j + \sum_i \sum_j p_{i,j} \ln p_{i,j} && (2.37)\\
&= \sum_i \sum_j p_{i,j} \left(\ln p_{i,j} - \ln p_i - \ln p_j\right) && (2.38)\\
&= \sum_{i,j} p_{i,j} \ln \frac{p_{i,j}}{p_i p_j} && (2.39)
\end{aligned}$$

Notice that if $\mathbf{f}$ and $\mathbf{g}$ are independent, then $p_{i,j} = p_i p_j$, and hence $I(\mathbf{f},\mathbf{g}) = 0$. This measure is maximized when the images are aligned (unlike entropy, which is minimized). That is, we wish to maximize the amount of information that is present in both images simultaneously. This makes sense since minimizing entropy, $H(\mathbf{f},\mathbf{g})$, maximizes its negative, $-H(\mathbf{f},\mathbf{g})$. The concept of mutual information was inherited from communications theory, where it was used to quantify the amount of information passed from a transmitter to a receiver [103].

Mutual information avoids the problem shown in Fig. 2.8(b) by including the marginal entropies of $\mathbf{f}$ and $\mathbf{g}$. Since the cost is only evaluated on the overlapping portion, including only background portions of $\mathbf{f}$ and $\mathbf{g}$ causes $H(\mathbf{f})$ and $H(\mathbf{g})$ to be very small, and the resulting mutual information will be small. Adding $H(\mathbf{f})$ and $H(\mathbf{g})$ to the equation tends to ensure that the imaged object remains in the region of overlap.

Still, there is a subtle problem with the mutual information measure. The motion correction algorithm might find a maximum by finding the transformation that minimizes the amount of blank overlap between the images [44, 86], as shown in Fig. 2.8(c). While

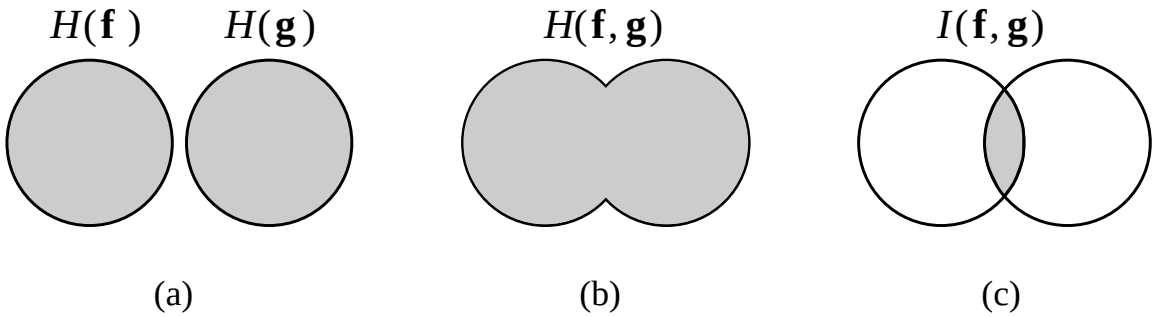


Figure 2.9: Three different entropy measures are shown as the shaded portions in (a), (b) and (c). The marginal entropies of $\mathbf{f}$ and $\mathbf{g}$ are given in (a). Joint entropy is shown in (b), and mutual information is shown in (c). (Adapted with permission from Studholme *et al.* [86])

the misalignment of the images increases $H(\mathbf{f}, \mathbf{g})$ (due to spreading of the joint histogram), including less image background increases both $H(\mathbf{f})$ and $H(\mathbf{g})$. It is possible that the increases in $H(\mathbf{f})$ and $H(\mathbf{g})$ are larger than the increase in $H(\mathbf{f}, \mathbf{g})$, yielding a maximum of the mutual information measure that does not correspond to the true alignment. To prevent this from happening, various forms of *normalized mutual information* (NMI) have been proposed [62, 86]. The objective of this set of cost functions is to compensate for the amount of background overlap. Common versions of this cost function take the form [46, 62, 86, 103]

$$NMI(\mathbf{f}, \mathbf{g}) = \frac{H(\mathbf{f}) + H(\mathbf{g})}{H(\mathbf{f}, \mathbf{g})} \ . \tag{2.40}$$

Notice that maximizing (2.40) is equivalent to maximizing

$$\frac{H(\mathbf{f}) + H(\mathbf{g}) - H(\mathbf{f}, \mathbf{g})}{H(\mathbf{f}, \mathbf{g})} \ , \tag{2.41}$$

which is simply the mutual information divided by the joint entropy.

Prior to the introduction of entropy-based measures into medical image registration, a similar measure that is often called the *correlation ratio* (CR) was proposed by Woods to register MRI and PET images [102]. The method is based on the same idea: that the mapping of intensities in the MR image to intensities in the PET image should be one-to-one, and that dispersion in the joint histogram indicates mis-alignment. However, rather than using an entropy measure, Woods calculated a normalized standard deviation for each MRI

intensity. That is, each intensity value in the MR image corresponds to a set of voxels. The intensities in the same set of voxels in the PET image are used to derive the cost function. In particular, the normalized standard deviation (NSD, standard deviation divided by the mean) of the intensities of the corresponding voxels in the PET image is calculated. The NSD's for all MRI intensities are combined, according to the intensity's weight in the MR image, to form the weighted mean of the normalized standard deviation. This quantity is minimized to achieve alignment.

Least Squares

Determining the patient's head motion from a time series of volumes is often posed as a least-squares optimization problem. It can be shown that this is the optimal cost function when two volumes differ only by Gaussian noise [44]. To solve the registration problem (see equation (2.19)) in the least-squares sense, we seek an $\mathbf{x}$ that minimizes the sum of squared residuals between $T(\mathbf{g}, \mathbf{x})$ and $\mathbf{f}$,

$$\min_{\mathbf{x}} \|T(\mathbf{g}, \mathbf{x}) - \mathbf{f}\| \ , \tag{2.42}$$

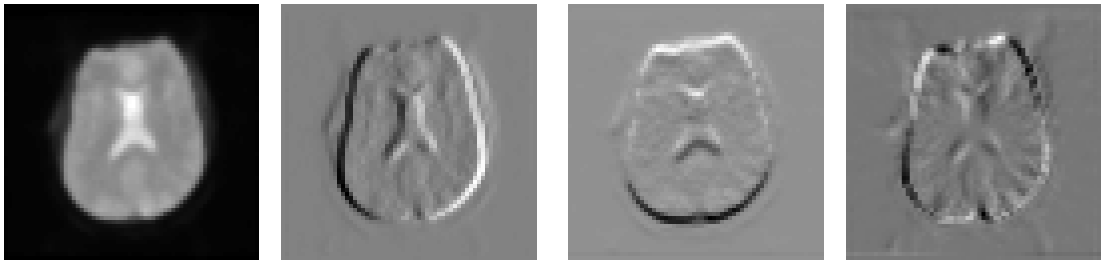
where $\|\cdot\|$ refers to the Frobenius norm. The word "residuals" refers to the voxel-by-voxel difference between the two volumes.

Rigid-body transformations are nonlinear and cannot be represented using a finite number of linear transformations. We can, however, linearize the operator by ignoring the higher-order terms in the Taylor expansion,

$$\begin{aligned} T(\mathbf{g}, \mathbf{x}) &= T(\mathbf{g}, 0) + \nabla T(\mathbf{g}, 0)\mathbf{x} + \mathcal{O}\left(\|\mathbf{x}\|^2\right) && (2.43) \\ &= \mathbf{g} + \mathbf{A}\mathbf{x} + \mathcal{O}\left(\|\mathbf{x}\|^2\right) \ . && (2.44) \end{aligned}$$

The matrix $\nabla T(\mathbf{g}, 0)$ (and thus $\mathbf{A}$) is an $m \times p$ matrix in which the i^{th} row holds the partial derivative of the intensity of the i^{th} voxel in $\mathbf{g}$ with respect to the p motion parameters. Equation (2.45) shows one row of $\mathbf{A}$ containing the six partial derivatives of $\mathbf{g}_i$ with respect to the three translations and three rotations of 3D rigid-body motion:

$$\mathbf{A} = \begin{bmatrix} \vdots \\ \frac{\partial \mathbf{g}_i}{\partial x} \frac{\partial \mathbf{g}_i}{\partial y} \frac{\partial \mathbf{g}_i}{\partial z} \frac{\partial \mathbf{g}_i}{\partial \theta_x} \frac{\partial \mathbf{g}_i}{\partial \theta_y} \frac{\partial \mathbf{g}_i}{\partial \theta_z} \\ \vdots \end{bmatrix} \ . \tag{2.45}$$



(a) $\mathbf{g}$ (b) $\frac{\partial \mathbf{g}}{\partial x}$ (c) $\frac{\partial \mathbf{g}}{\partial y}$ (d) $\frac{\partial \mathbf{g}}{\partial \theta_z}$

Figure 2.10: Slice 16 of the base volume $\mathbf{g}$, as well as its partial derivatives with respect to x translation, y translation, and rotation about the x-axis. The derivative images were created using the finite difference method outlined in (2.46).

Alternatively, one can think of $\mathbf{A}$ as containing six columns. Each column of $\mathbf{A}$ is the partial derivative of $\mathbf{g}$ with respect to one of the motion parameters. These columns (volumes) look like the derivative images of $\mathbf{g}$, as shown in Fig. 2.10. The partial derivatives can be estimated using a finite difference approximation. For example, $\mathbf{g}$ can be rotated about the z-axis by $0.01°$ and $-0.01°$ to yield the volumes $\mathbf{g}^+$ and $\mathbf{g}^-$, respectively. Then, the partial derivative of voxel i with respect to rotation about the z-axis can be approximated using the finite difference formula

$$\frac{\partial \mathbf{g}_i}{\partial \theta_z} \approx \frac{\mathbf{g}_i^+ - \mathbf{g}_i^-}{0.02} \ . \tag{2.46}$$

The vector $\mathbf{x}$ in (2.43) holds the linear motion parameters. By substituting the approximation for T into (2.42), we get

$$\min_{\mathbf{x}} \|\mathbf{g} + \mathbf{A}\mathbf{x} - \mathbf{f}\| \ . \tag{2.47}$$

Equation (2.47) is a linear least-squares problem. Assuming that $\mathbf{A}$ is of full rank, we can attain an analytical solution [12, 39] using matrix pseudo-inversion

$$\begin{aligned} \mathbf{g} + \mathbf{A}\mathbf{x} &= \mathbf{f} && (2.48)\\ \mathbf{A}\mathbf{x} &= \mathbf{f} - \mathbf{g} && (2.49)\\ \mathbf{A}'\mathbf{A}\mathbf{x} &= \mathbf{A}'(\mathbf{f} - \mathbf{g}) && (2.50)\\ \mathbf{x} &= \left(\mathbf{A}'\mathbf{A}\right)^{-1}\mathbf{A}'(\mathbf{f} - \mathbf{g}) \ . && (2.51) \end{aligned}$$

Similar to $\mathbf{B}^\dagger$ in secion 2.2.3, we denote $\left(\mathbf{A}'\mathbf{A}\right)^{-1}\mathbf{A}'$, the matrix pseudo-inverse of $\mathbf{A}$, with

$\mathbf{A}^\dagger$. Using this notation, the least-squares solution to (2.47) can be written

$$\mathbf{x} = \mathbf{A}^\dagger (\mathbf{f} - \mathbf{g}) \ . \tag{2.52}$$

After finding $\mathbf{x}$, the floating volume can be resampled using the inverse of T, and another motion estimate can be obtained from the new dataset. By applying this approximation iteratively, we generate a sequence of linear motion estimates, $\mathbf{x}^{(k)}$. These increments are accumulated into a current best-guess set of motion parameters by simple addition. If this best-guess is denoted $\Sigma\mathbf{x}^{(k)}$, then we can represent the k^{th} resampled volume as a transformed version of the original floating volume,

$$\mathbf{f}^{(k)} = T^{-1}(\mathbf{f}^{(0)}, \Sigma\mathbf{x}^{(k)}) = T^{-1}(\mathbf{f}, \Sigma\mathbf{x}^{(k)}) \ , \tag{2.53}$$

where $\mathbf{f}^{(0)}$ is defined as $\mathbf{f}$. Under certain mild conditions, the motion increments will converge as each iteration brings $T^{-1}(\mathbf{f}, \Sigma\mathbf{x}^{(k)})$ closer to alignment with $\mathbf{g}$. This technique is called *fixed point iteration*.

The issue of resampling deserves some attention here. Resampling causes errors in the data, a result of interpolating at points where we do not actually have a sample. Furthermore, sequential resampling (resampling the resampled dataset) causes the errors to accumulate. Clearly, it is most accurate to resample as infrequently as possible. In most implementations of the above iterative least-squares optimization, the sequence of $\mathbf{x}$'s are composed into a single rigid-body transform. This transform is applied directly to the original floating volume to produce the next $\mathbf{f}$ for use in (2.52). Hence, the algorithm can be designed to avoid resampling volumes that have already been resampled.

In fMRI, this type of registration is performed on a time series of n volumes, aligning each with a chosen base volume. The above notation is general enough to represent the registration of all volumes at once with a single matrix expression. Again, we use the dataset matrix formulation outlined in section 2.2.3, in which $\mathbf{F}$ has n columns, each holding a volume. Since $\mathbf{F}$ has n columns, the motion parameter matrix must also have n columns, one set of motion parameters for each volume in the time series. Hence, we define $\mathbf{X}$ as just such a matrix. We must also define $\mathbf{G}$ as a matrix that holds a copy of the reference volume in each of its n columns. Thus, the full time-series registration problem is depicted by the expression

$$\min_{\mathbf{X}} \|\mathbf{G} + \mathbf{A}\mathbf{X} - \mathbf{F}\| \ . \tag{2.54}$$

If $T^{-1}(\mathbf{F}, \Sigma\mathbf{X}^{(k)})$ represents the resampled dataset in which the n cumulative rigid-body transformations held in $\Sigma\mathbf{X}^{(k)}$ are applied to the columns of $\mathbf{F}$, then our n-volume registration problem can be solved by iterating the system

$$\begin{cases} \mathbf{F}^{(k)} = T^{-1}\left(\mathbf{F}^{(0)}, \Sigma\mathbf{X}^{(k-1)}\right) \\ \mathbf{X}^{(k)} = \mathbf{A}^{\dagger}\left(\mathbf{F}^{(k)} - \mathbf{G}\right) \end{cases} \tag{2.55}$$

until the motion increments are sufficiently small.

The least-squares cost function has been widely used to correct for intra-scan motion artifacts [5, 73, 83, 91, 95]. Multiple images can be combined and compared using the least-squares cost function to generate a single, better resolved image [5, 95]. The cost function can also be applied to navigator echoes to determine the best fit between the echoes [73, 83, 91].

Least squares is also used frequently to correct inter-scan motion. Many implementations use the Levenberg-Marquardt optimization scheme, which utilizes a second-degree Taylor approximation to the rigid-body transformation [21, 42, 66, 87, 101]. While the convergence can be faster, estimating the second derivative is expensive, and the results depend on the stability of the approximation. There are other methods used to minimize the sum of squares. For example, simplex-based optimization methods have also been used [23, 96, 97]. Also, if the optimization involves only one variable, then the optimal solution can feasibly be found simply by sampling the sum of squares cost function at different locations. This method is often used in conjunction with navigator techniques [56, 55, 93].

The least-squares cost function makes the assumption that corresponding voxels in $\mathbf{f}$ and $\mathbf{g}$ should have the same intensity once they are correctly aligned. This may not always be the case, since noise is often contained in the images. However, unlike mutual information, images involving a highly nonlinear transfer function (from the intensities in one image to the intensities in the other) likely will not minimize the least-squares cost function even when the images are perfectly aligned.

However, global intensity scaling can be incorporated into the least-squares cost function, yielding what is called the *scaled least-squares* cost function, defined as

$$S(\mathbf{f}, \mathbf{g}) = \sum_i \left(\mathbf{f}_i - \nu\mathbf{g}_i\right)^2 \ , \tag{2.56}$$

where ν is an arbitrary, non-zero, constant scaling factor. Since the action of scaling is linear, it can easily be included in the least-squares optimization above by adding one

more parameter, global scaling, to the current list of six. We will distinguish between the two vectors of motion parameters by referring to the six-parameter vector as $\mathbf{x}$ and the augmented, seven-parameter vector as $\mathcal{X}$, which equals $[\mathbf{x}' \ \nu]'$. When calculating the gradient matrix $\nabla T(\mathbf{g}, \mathcal{X})$, the seventh column holds the derivative of the voxel intensities with respect to this scaling factor. That is, if ν is the scaling factor, and $\mathbf{g}_i$ is the voxel intensity before the scaling, then $\nu\mathbf{g}_i$ is the voxel intensity after the scaling. The derivative of $\nu\mathbf{g}_i$ with respect to ν is simply $\mathbf{g}_i$. Hence, the seventh column of $\nabla T(\mathbf{g}, \mathcal{X})$ holds a copy of $\mathbf{g}$. We will denote the six-column and seven-column matrices of partial derivatives as $\mathbf{A}$ and $\mathbf{A}_7$, respectively.

Since our base volume can be accounted for by letting ν equal 1, we do not need to explicitly include $\mathbf{g}$ in the model. That is, if ν is one, we get $\mathbf{g}+\mathbf{A}\mathbf{x} = \mathbf{A}_7\mathcal{X}$. Hence, our linear approximation to the rigid-body transformation is simply $T(\mathbf{g}, \mathcal{X}) = \mathbf{A}_7\mathcal{X}$. Then, (2.48) and its solution can be restated as

$$\mathbf{A}_7\mathcal{X} = \mathbf{f} \tag{2.57}$$

$$\mathcal{X} = \mathbf{A}_7^{\dagger}\mathbf{f}. \tag{2.58}$$

Using this method, all six original motion parameters, as well as the global scaling factor, are determined simultaneously in an effort to minimize the corresponding least-squares cost function.

Finally, the contribution of each voxel to the sum of squares cost function can be varied by including a weighting matrix, $\mathbf{W}$, in the calculation [73, 95]. The matrix $\mathbf{W}$ is a square matrix that has dimensions $N \times N$, where N is the number of voxels in the volumes being compared. Essentially, $\mathbf{W}$ acts to scale the residual of each voxel, allowing a different scale factor to be applied to each and every voxel. This enables one to selectively assign significance to various parts of the image by varying the weights. To emphasize the importance of a particular voxel's fit, that voxel's weight can be increase. This weighting changes the pseudo-inverse solution from that stated in (2.52) to

$$\mathbf{x} = \left(\mathbf{A}'\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}'\mathbf{W}(\mathbf{f}-\mathbf{g}) \ . \tag{2.59}$$

The sum of squares cost function is noted for its sensitivity to outliers. Since the cost function depends on the square of the residual, a single voxel with a large residual can influence the error a great deal.

Robust Estimators

When two images are subtracted, it is often informative to consider the distribution of the resulting residuals. If the residuals are expected to have a Gaussian distribution, then the least-squares cost function is an effective measure of dissimilarity for registration purposes [44]. However, the distribution may very well contain significant outliers that invalidate the Gaussian noise model. It is often desirable to de-emphasize the effect of these outliers, and use the remaining Gaussian-like residuals to evaluate the cost function. This behaviour can be built into the cost function, and these types of cost functions are called *robust estimators*.

Unlike the least-squares cost function, the cost of a robust estimator function is bounded. As the residuals get larger, the robust estimator cost function levels off and saturates so that an increase in residual causes only a marginal (or zero) increase in cost.

There are many different robust estimator cost functions, all having the same general shape. A common one is the Geman-McClure estimator

$$GM(\mathbf{f}, \mathbf{g}; C) = \sum_i \rho(\mathbf{f}_i - \mathbf{g}_i, C) \ , \tag{2.60}$$

where

$$\rho(r, C) = \frac{r^2}{C^2 + r^2} \ . \tag{2.61}$$

A plot of this error function is shown in Fig. 2.11.

Robust estimators were used by Nikou *et al.* [74] for both intra-modality and inter-modality medical volume registration. The study found that robust estimator cost functions were able to register multiple volumes despite small physiological differences, such as increasing severity of multiple sclerosis. Nestares *et al.* [72] also proposed the use of robust estimators to align multiple MRI volumes acquired using different scanning protocols.

Ratio Image Uniformity

The ratio image uniformity (RIU) cost function is based on the assumption that the two images being aligned differ only by mis-alignment, and by a global intensity scaling. Hence, when the two images are perfectly aligned, dividing one by the other on a voxel-by-voxel basis should yield a ratio image that is simply a constant value; the whole image is a single shade of gray (ignoring division-by-zero issues). Consider a comparison between volumes $\mathbf{f}$ and $\mathbf{g}$. For voxel i, dividing $\mathbf{f}_i$ by $\mathbf{g}_i$ gives their ratio, $\mathbf{z}_i$. The ratio image uniformity

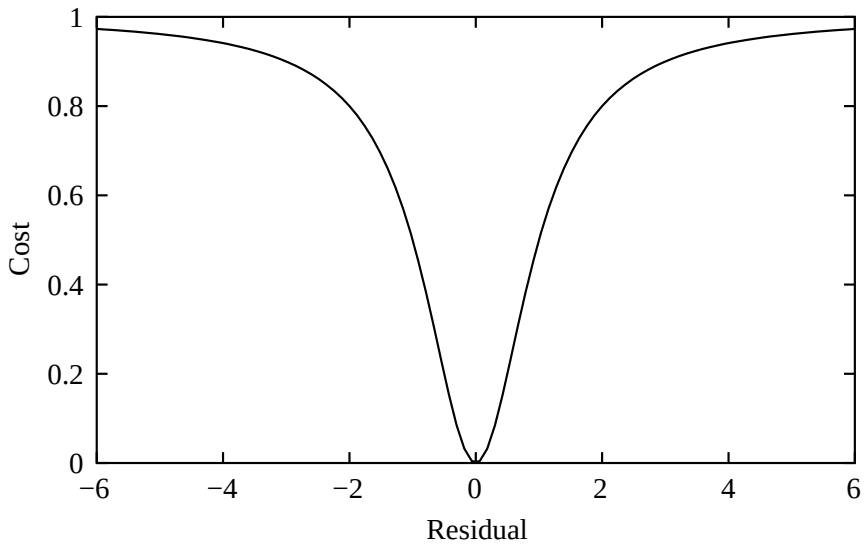


Figure 2.11: Geman-McClure robust estimator cost function for $C = 1$.

measure is the normalized standard deviation (standard deviation divided by the mean) of the ratio image, $\mathbf{z}$. The formula is

$$RIU(\mathbf{f}, \mathbf{g}) = \frac{1}{N} \frac{\sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}})^2}{\bar{\mathbf{z}}} \ , \tag{2.62}$$

where

$$\mathbf{z}_i = \frac{\mathbf{f}_i}{\mathbf{g}_i} \quad \text{and} \quad \bar{\mathbf{z}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \ . \tag{2.63}$$

Ratio image uniformity has been shown to be effective for registering PET images [100, 101, 103]. However, when registering MR images, the performance of RIU is generally considered to be weaker than that of mutual information and scaled least-squares, although the results are somewhat inconclusive due to other factors [46, 85, 100, 101, 103].

A modified version of the RIU cost function has been proposed by Holden *et al.* [46]. The intensity ratio for voxels of low intensity cause the measure to be highly sensitive to noise. When comparing an image to a noisy copy of itself, a background voxel with low intensity might have a value of 1 in one image, and a value of 5 in the noisy image. Thus, the ratio is 5 (or 0.2). The same noise, applied to a voxel with intensity 100, would correspond to a voxel with intensity 104, and a ratio of 1.04 (or 0.9615). For these two images, their nominal scaling factor should be close to 1. Thus, the low-intensity voxels exhibit much higher variability, disrupting the ratio image uniformity. Holden proposed a similar cost function

by adding a constant value, c, to the intensity of all voxels in both images. Moreover, he suggested that choosing c to be the maximum intensity from the reference volume works well. The formula for the *modified ratio image uniformity* (MRIU) cost function is the same as in equations (2.62)-(2.63) except that $\mathbf{z}_i$ is defined as $\mathbf{z}_i = \frac{\mathbf{f}_i + c}{\mathbf{g}_i + c}$. The MRIU cost function was found to be more consistent than RIU when registering MR images [46].

Chapter 3

Simultaneous Solution

As was outlined in section 1.4, the presence of BOLD activation in fMRI datasets causes errors in the resulting motion parameters because the voxels that are active in one volume, but not in the other, are treated as outliers when the cost function is evaluated. This error was demonstrated for the least-squares cost function. However, many other cost functions exhibit the same behaviour.

A study that compared the influence of the activation on different registration cost functions was conducted by Freire *et al.* [33]. In that study, the least-squares cost function (as well as the correlation ratio cost function mentioned in section 2.3.3 on page 43) was again shown to generate stimulus-correlated motion estimates. The motion estimates generated by the ratio image uniformity cost function did not exhibit obvious activation bias because the noise present in the estimates was considerably larger than that of the other methods. Ratio image uniformity was designed for inter-modality registration, and is generally not considered to have appropriate accuracy for fMRI datasets.

The Geman-McClure robust estimator cost function was tuned to exhibit relatively little activation bias. However, despite implementing a multi-resolution optimization scheme, the GM cost function was found to trade off activation sensitivity with susceptibility to local minima. For larger C-values (see equations (2.60) and (2.61)), motion estimates exhibited activation bias, while for lower C-values, motion estimates were more noisy. This behaviour stems from the fact that a low C-value causes a large portion of the residuals to fall into the tails of the cost function and have little influence on the cost, leaving a smaller portion of the voxels to guide the registration process. A larger C-value makes the GM cost function behave more like a least-squares cost function.

Mutual information was deemed the most appropriate cost function to register high-field fMRI datasets even though MI also exhibited stimulus-correlated motion errors, but with a small amplitude. However, it is difficult to draw any meaningful conclusions from the study because some of the registration methods were run on spatially smoothed datasets, while others were run on datasets that were not smoothed.

While some cost functions seem less prone to activation bias than others, none of them specifically address the interplay between motion and activation. Motion should be removed from the dataset before activation detection is performed. However, activation needs to be removed from the dataset before a reliable estimate of the motion can be attained. This circular argument suggests that the two problems, registration and activation detection, are coupled and should be solved simultaneously. In this chapter, we combine these two problems and investigate various simultaneous solution strategies.

3.1 Combining Registration and Activation Detection

We propose a new approach, called the *Simultaneous Registration and Activation* (SRA) algorithm, that avoids the interference of the BOLD signal in the least-squares registration algorithm. Recall from section 2.2.3 that the activation detection least-squares problem can be stated as

$$\min_{\mathbf{Y}} \|\mathbf{G} + \mathbf{YB} - \mathbf{F}\| \ . \tag{3.1}$$

Also, recall from section 2.3.3 that the registration least-squares problem can be stated as

$$\min_{\mathbf{X}} \|\mathbf{G} + \mathbf{AX} - \mathbf{F}\| \ . \tag{3.2}$$

These two problems have a similar formulation. By combining them, a model to account for both motion and activation can be stated in a single matrix expression:

$$\{\texttt{dataset}\} \approx \{\texttt{baseline}\} + \{\texttt{motion}\} + \{\texttt{activation}\} \tag{3.3}$$

$$\mathbf{F} \approx \mathbf{G} + \mathbf{AX} + \mathbf{YB} \ . \tag{3.4}$$

Recall that $\mathbf{F}$ holds a full volume in each column. For a dataset of volumes with dimensions $64 \times 64 \times 30$, the matrices $\mathbf{F}$, $\mathbf{G}$, $\mathbf{A}$ and $\mathbf{Y}$ all have 122,880 rows each. Hence, columns represent the spatial component of the dataset. On the other hand, rows represent the temporal component of the dataset. The matrices $\mathbf{F}$, $\mathbf{G}$, $\mathbf{X}$ and $\mathbf{B}$ all hold time-series

in their rows. For a dataset with 40 volumes, they each have 40 columns. If we are using six-degree-of-freedom motion and two stimulus regressors, then our matrix dimensions are

$$\mathbf{F} \approx \mathbf{G} + \mathbf{AX} + \mathbf{YB} \tag{3.5}$$

$$[122,880 \times 40] \approx [122,880 \times 40] + [122,880 \times 6]\,[6 \times 40] + [122,880 \times 2]\,[2 \times 40] \tag{3.6}$$

Our new least-squares problem formulation then becomes

$$\min_{(\mathbf{X},\mathbf{Y})} \|\mathbf{AX} + \mathbf{YB} - \mathbf{C}\| \ , \tag{3.7}$$

where $\|\cdot\|$ again refers to the Frobenius norm, and we have replaced $(\mathbf{F} - \mathbf{G})$ with $\mathbf{C}$.

Since $\mathbf{G} + \mathbf{AX}$ is only an approximation to the rigid-body transformation, (3.7) is also an approximation, and must be solved iteratively just like in the original registration problem. The advantage to this formulation is that the effects of both motion and activation can be accounted for. Hence, we can solve the least-squares problem for $\mathbf{X}$ and $\mathbf{Y}$ simultaneously. The solution gives the rigid-body motion increment ($\mathbf{X}$) and activation ($\mathbf{Y}$) that minimizes the sum of squared residuals between $\mathbf{F}$ and $\mathbf{G}$, accounting for both motion and BOLD contrast in parallel.

3.2 General Solution

The least-squares solution to (3.7) is not unique. To demonstrate this fact, let $(\mathbf{X}, \mathbf{Y})$ be a solution to (3.7), and consider the perturbation $(\mathbf{X} + \delta_{\mathbf{X}}, \mathbf{Y} + \delta_{\mathbf{Y}})$ in the following derivation:

$$\|\mathbf{A}(\mathbf{X} + \delta_{\mathbf{X}}) + (\mathbf{Y} + \delta_{\mathbf{Y}})\mathbf{B} - \mathbf{C}\| \tag{3.8}$$

$$= \|\mathbf{AX} + \mathbf{YB} - \mathbf{C} + \mathbf{A}\delta_{\mathbf{X}} + \delta_{\mathbf{Y}}\mathbf{B}\| \ . \tag{3.9}$$

Thus, if we have

$$\mathbf{A}\delta_{\mathbf{X}} = -\delta_{\mathbf{Y}}\mathbf{B} \ , \tag{3.10}$$

then the perturbed solution $(\mathbf{X} + \delta_{\mathbf{X}}, \mathbf{Y} + \delta_{\mathbf{Y}})$ has the same cost as the solution $(\mathbf{X}, \mathbf{Y})$, and is thus an equivalent solution. Moreover, we can show that for (3.10) to hold for arbitrary $\mathbf{A}$ and $\mathbf{B}$, $\delta_{\mathbf{X}}$ must be $\alpha\mathbf{B}$ and $\delta_{\mathbf{Y}}$ must be $-\mathbf{A}\alpha$, where α is an arbitrary $p \times s$ matrix. Recall that $\mathbf{A}$ is $p \times m$, where $m > p$, and $\mathbf{B}$ is $s \times n$, where $n > s$. The right-hand-side of (3.10) is a linear combination of the rows of $\mathbf{B}$. Hence, the left-hand-side must be in the row-span of $\mathbf{B}$ and, without loss of generality, we can represent $\delta_{\mathbf{X}}$ as $\alpha_1\mathbf{B}$ where α_1 is arbitrary. By

a similar argument, we can conclude that setting $\delta_{\mathbf{Y}}$ equal to $-\mathbf{A}\alpha_2$ captures all possible options for $\delta_{\mathbf{Y}}$. Since we now have $\mathbf{A}\alpha_1\mathbf{B} = \mathbf{A}\alpha_2\mathbf{B}$, and assuming that $\mathbf{A}$ and $\mathbf{B}$ have full rank, we arrive at $\alpha = \alpha_1 = \alpha_2$. Hence, the solution space to (3.7) is the (ps)-dimensional space

$$\Big\{(\mathbf{X} + \alpha\mathbf{B}, \mathbf{Y} - \mathbf{A}\alpha) \mid \forall\alpha\Big\}, \tag{3.11}$$

where $(\mathbf{X}, \mathbf{Y})$ is any particular solution. This is true for any single iteration of the fixed point iteration.

Another interesting and intuitive way to see how this non-uniqueness allows for interchange between the registration and activation detection problems is to consider the following derivation:

$$\begin{aligned} & \mathbf{A}(\mathbf{X} + \alpha\mathbf{B}) + \mathbf{Y}\mathbf{B} - \mathbf{C} && (3.12)\\ = \quad & \mathbf{A}\mathbf{X} + \mathbf{A}\alpha\mathbf{B} + \mathbf{Y}\mathbf{B} - \mathbf{C} && (3.13)\\ = \quad & \mathbf{A}\mathbf{X} + (\mathbf{Y} + \mathbf{A}\alpha)\mathbf{B} - \mathbf{C}\ . && (3.14) \end{aligned}$$

In other words, dataset variation due to a small stimulus-correlated motion can be represented as a small brain-edge artifact in the activation map, and vice versa.

When an optimization problem does not have a unique solution or is ill-posed, a common strategy used to steer the problem toward a sensible, unique solution is to apply regularization. This can be accomplished by reformulating (3.7) to include an additional parameter-dependent term, yielding the regularized problem

$$\min_{(\mathbf{X},\mathbf{Y})} \Big[\ \|\mathbf{A}\mathbf{X} + \mathbf{Y}\mathbf{B} - \mathbf{C}\| + \gamma\Phi(\mathbf{X}, \mathbf{Y})\ \Big]\ . \tag{3.15}$$

The regularization term, $\gamma\Phi(\mathbf{X}, \mathbf{Y})$, is chosen to alter the cost function in a way that steers the optimization toward a solution which achieves a compromise between minimizing the original cost function and minimizing the function Φ. The relative weighting of the original cost function and the regularization term is controlled by the regularization parameter, γ. One of the challenges for this type of formulation is choosing an appropriate value for γ (see [43] for details). Once a value is chosen, there are numerical methods (depending on the form of Φ) that solve such optimization problems [1, 50, 59].

In our case, however, we already know that (3.11) is a parameterization of the solution space of (3.7). Solving the constrained optimization problem can then be accomplished with a two-step process. First, we find a particular solution, $(\mathbf{X}, \mathbf{Y})$, and use it as the anchor

of our parameterized general solution. Then, we use an additional constraint to find the α that gives us our desired, unique solution.

3.3 Particular Solution

In order to obtain the general solution stated in (3.11), we must first find a particular solution to (3.7). We start the derivation by finding the QR decomposition (see [39], section 5.2) of $\mathbf{B}'$,

$$\mathbf{B}' = [\mathbf{Q}_1\ \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix} \Leftrightarrow \mathbf{B} = [\mathbf{R}'\ 0] \begin{bmatrix} \mathbf{Q}_1' \\ \mathbf{Q}_2' \end{bmatrix} , \tag{3.16}$$

where $[\mathbf{Q}_1\ \mathbf{Q}_2]$ and $[\mathbf{R}'\ 0]$ are block matrices such that $\mathbf{R}$ has dimensions $s \times s$ and is upper-triangular. Equation (3.7) can then be rewritten

$$\min_{(\mathbf{X},\mathbf{Y})} \left\| \mathbf{AX} + \mathbf{Y} \left[\mathbf{R}'\ 0\right] \begin{bmatrix} \mathbf{Q}_1' \\ \mathbf{Q}_2' \end{bmatrix} - \mathbf{C} \right\| \tag{3.17}$$

$$= \min_{(\mathbf{X},\mathbf{Y})} \|\mathbf{AX}[\mathbf{Q}_1\ \mathbf{Q}_2] + \mathbf{Y}[\mathbf{R}'\ 0] - \mathbf{C}[\mathbf{Q}_1\ \mathbf{Q}_2]\| \tag{3.18}$$

$$= \min_{(\mathbf{X},\mathbf{Y})} \|\mathbf{AX}[0\ \mathbf{Q}_2] + \mathbf{AX}[\mathbf{Q}_1\ 0] + \mathbf{Y}[\mathbf{R}'\ 0] - \mathbf{C}[\mathbf{Q}_1\ 0] - \mathbf{C}[0\ \mathbf{Q}_2]\| . \tag{3.19}$$

The fact that (3.17) is equal to (3.18) comes from the fact that the matrix $[\mathbf{Q}_1\ \mathbf{Q}_2]$ is orthogonal, and that the Frobenius norm is invariant under multiplication by an orthogonal (or "unitary") matrix [39]. For any matrix $\mathbf{M}$, and appropriately sized orthogonal matrix $\mathbf{Q}$, we have the identity

$$\|\mathbf{MQ}\| = \|\mathbf{M}\| , \tag{3.20}$$

where (as before) $\|\cdot\|$ is the Frobenius norm.

At this intermediate step, it should be noted that an equivalent approach is to take the QR decomposition of $\mathbf{A}$. However, $\mathbf{A}$ is much larger than $\mathbf{B}$. For a dataset containing volumes with dimensions $64 \times 64 \times 30$, $\mathbf{A}$ has 122,880 rows, one row for each voxel. The resulting $\mathbf{Q}$ matrix contains over 15 million elements, and requires over 57 gigabytes of memory to store using single-precision floating point representation. The matrix $\mathbf{B}'$, on the other hand, has one row for every time step. For a dataset containing 200 time steps, the corresponding $\mathbf{Q}$ matrix requires only 156 kB of memory. Very few fMRI experiments contain more than 200 time steps.

Multiplying a row-vector on the right by $[\mathbf{Q}_1\ \mathbf{Q}_2]$ is essentially a coordinate transformation to a reference-frame for which the columns of $[\mathbf{Q}_1\ \mathbf{Q}_2]$ form the basis. The matrix

$\mathbf{Q}_1$ has s columns spanning the row-space of $\mathbf{B}$ (which is the column-space of $\mathbf{B}'$). The remaining columns, stored in $\mathbf{Q}_2$, span the orthogonal space. Thus, the matrix $\mathbf{XQ}_1$ corresponds to the components of $\mathbf{X}$ that are in the row-space of $\mathbf{B}$. Since our goal is to find *any* particular solution to (3.7), these components can be set to zero without loss of generality because our general solution in (3.11) already includes the row-space of $\mathbf{B}$. Setting these elements to zero simply forces the resulting rows in our $\mathbf{X}$ solution to be orthogonal to all the stimulus regressors in the rows of $\mathbf{B}$.

Thus, to arrive at an initial, unconstrained solution, we can assume that the term $\mathbf{AX}[\mathbf{Q}_1\ 0]$ from (3.19) is zero. Equation (3.19) then becomes

$$\min_{(\mathbf{X},\mathbf{Y})} \|\mathbf{AX}[0\ \mathbf{Q}_2] + \mathbf{Y}[\mathbf{R}'\ 0] - \mathbf{C}[\mathbf{Q}_1\ 0] - \mathbf{C}[0\ \mathbf{Q}_2]\| \ , \tag{3.21}$$

and our variables $\mathbf{X}$ and $\mathbf{Y}$ are decoupled. This allows us to express the combined least-squares problem as the sum of two independent least-squares problems, yielding

$$\min_{\mathbf{X}} \|\mathbf{AX}[0\ \mathbf{Q}_2] - \mathbf{C}[0\ \mathbf{Q}_2]\| \quad + \quad \min_{\mathbf{Y}} \|\mathbf{Y}[\mathbf{R}'\ 0] - \mathbf{C}[\mathbf{Q}_1\ 0]\| \ . \tag{3.22}$$

Assuming $\mathbf{R}$ is non-singular (which must be the case if all the regressors in $\mathbf{B}$ are linearly independent) and that $\mathbf{A}$ has full column rank, the system has the solution

$$\mathbf{X} = \mathbf{A}^{\dagger}\mathbf{C}\mathbf{Q}_2\mathbf{Q}_2' \tag{3.23}$$

$$\mathbf{Y} = \mathbf{C}\mathbf{Q}_1(\mathbf{R}')^{-1} \ . \tag{3.24}$$

3.4 Constrained Solution

Equations (3.23) and (3.24) yield a particular solution to the combined least-squares problem stated in (3.7). This particular solution is used in (3.11) to get the general solution. The task remains to determine which solution in the multi-dimensional solution space is appropriate for our needs. We need a mathematical expression that distinguishes one solution from all the others in the solution space. That is, what additional constraint can we impose on our general solution $(\mathbf{X} + \alpha\mathbf{B}, \mathbf{Y} - \mathbf{A}\alpha)$ to obtain a solution that is meaningful in the context of fMRI?

The approach we take is to use the fact that the activation associated with any given stimulus is localized, and typically covers less than 15% of the brain. Hence, we would expect only small clusters of elements in the activation map (the active regions) to have relatively

high intensities, and all other elements to have intensities close to zero (the inactive regions). We will refer to this characteristic as *sparsity*.

Any particular solution we obtain for $\mathbf{Y}$ is composed of the true, desired activation mixed with the columns of $\mathbf{A}$. Moreover, other equivalent solutions can be derived from the particular solution by adding linear combinations of the columns of $\mathbf{A}$.

Recall that each column of $\mathbf{A}$ holds the partial derivative of $\mathbf{g}$ with respect to a motion parameter, and is generally non-zero for extended regions of the brain. Thus, adding linear combinations of the columns of $\mathbf{A}$ to the true activation will cause large non-zero regions throughout the brain, which is counter to the sparsity condition. To isolate the true activation, we seek an α that removes these $\mathbf{A}$ components from our particular solution, causing the majority of the elements of $(\mathbf{Y} - \mathbf{A}\alpha)$ to be close to zero. This result can be largely achieved by imposing the constraint

$$\min_{\alpha} \sum_{i} \arctan\left(c|\mathbf{Y}_i - [\mathbf{A}\alpha]_i\,|\right) \ , \tag{3.25}$$

where $[\mathbf{A}\alpha]_i$ refers to the i^{th} voxel in the column vector $\mathbf{A}\alpha$, and c is a chosen constant. This cost function is similar in nature to the Geman-McClure robust estimator [74] discussed in section 2.3.3 (on page 49) since the arctan function is bounded above by a horizontal asymptote. Further discussion on the behaviour and performance of this and other constraints is presented in section 4.3.

3.5 Solution by Paired Iteration

Without the above analysis that formulates and solves the combined problem, a natural approach to overcome activation bias is to iterate the two processing steps with the hope that the final result will reflect both a minimum in the registration problem and a minimum in the activation problem. For example, one could perform motion correction first, and then run activation detection on the "aligned" dataset. After removing the estimated activation, the process could be repeated until subsequent iterations yield no change and the iterative method converges. This is a form of fixed-point iteration, which we will refer to as *paired iteration*. Unfortunately, this method rarely yields the expected motion parameters and activation maps. Instead, the activation maps often contain brain-edge artifacts, and the motion parameters often contain large stimulus-correlated components.

A variation of this approach has been proposed by Freire *et al.* [32]. In this two-stage approach, the initial registration step is followed by activation detection. The resulting activation map is thresholded and dilated to form a binary activation mask. Voxels under the mask are considered to be active and are excluded from a second registration step, thereby shifting all the weight of the registration process onto the unmasked portion of the volume. While this method has shown good results on both simulated and *in vivo* datasets, the author points out that fMRI experiments that contain large regions of activation may cause the method to be less accurate, since the registration must then rely on information from fewer voxels. Also, the binary inclusion/exclusion of voxels may cause the method to be unstable. Furthermore, this heuristic two-stage method offers no guarantees about the nature of the solution. The SRA method, on the other hand, yields a solution that minimizes the sum of squared residuals over the possible motion parameters and activation maps.

While the SRA method outlined in section 3.3 finds a solution to the combined least-squares problem, it requires a custom implementation. The paired iteration approach described above can yield exactly the same results as the SRA method by introducing a simple adjustment step. We have developed an algorithm that uses almost entirely "off-the-shelf" software to accurately resolve the motion/activation interdependency issue. Furthermore, this work introduces a framework by which this interdependency can be analyzed.

It should be noted that the paired iteration method creates the possibility for two levels of iteration. First, the paired iteration method is iterative by nature, and we will refer to its iteration as the *outer loop*. In each iteration of the outer loop, the registration program is executed. This registration program may have its own built-in iterative method. We will refer to this built-in iteration as the *inner loop*. It is not necessary for the inner loop to converge during each iteration of the outer loop. In fact, the convergence criteria can be removed from the inner loop altogether, and replaced by a convergence test in the outer loop. Thus, the function of the registration method is simply to supply an increment to the motion parameters that will bring the outer loop closer to convergence. Many registration programs allow the user to specify the maximum number of internal iterations. Setting that maximum to one is sufficient.

3.5.1 Constrained Paired Iteration

It may come as a surprise to learn that the paired iteration method can be used to produce the same solution as the SRA method. In fact, we will even show that a version of the paired iteration method is mathematically equivalent to SRA.

Consider the result of running the paired iteration method to convergence. While the final solution is a least-squares solution, it is likely not appropriate to the problem context. We could apply the constraint in (3.25), but if the resulting change in $\mathbf{X}$ is not sufficiently small, the linear approximation to the rigid-body transformation will be less accurate, forcing us to iterate further.

The safest approach is to apply the constraint after each iteration. In this way, any least-squares solution that is converged to will also satisfy (3.25). Hence, we introduce the method of *constrained paired iteration* (CPI). Figure 3.1 depicts two versions of the CPI method in data flowcharts. In Fig. 3.1(a), the activation detection is performed first, followed by registration. We will refer to this method as the CPI-A method, the "A" representing the fact that activation detection is performed first. The resulting $\mathbf{X}$ and $\mathbf{Y}$ estimates are then fed into the contraint (as well as the matrices $\mathbf{A}$ and $\mathbf{B}$). The adjusted motion estimates, $\tilde{\mathbf{X}}$, are then combined with the current best-guess set of motion parameters (initially set to zero), and used to resample the original dataset.

Another approach to the paired iteration method is to start the process with registration, followed by activation detection, as shown in Fig. 3.1(b). This method will be referred to as the CPI-R method, the "R" standing for the fact that registration is performed first. Again, we obtain initial estimates for $\mathbf{X}$ and $\mathbf{Y}$. As above, the constraint is applied, and the solutions for $\mathbf{X}$ and $\mathbf{Y}$ are adjusted. The new $\tilde{\mathbf{X}}$ is used to update the current best-guess set of motion parameters, and the original dataset is resampled.

For comparison purposes, the *unconstrained paired iteration* (UPI) method will refer to the paired iteration method that does not use the additional constraint. Just like with the CPI method, we will use either an "A" or an "R" suffix to indicate whether activation detection or registration is done first.

3.5.2 Equivalence to the SRA Method

The CPI-A method is mathematically equivalent to the SRA method. That is, we will show that each iteration of the CPI-A method produces the same solution as each iteration of the

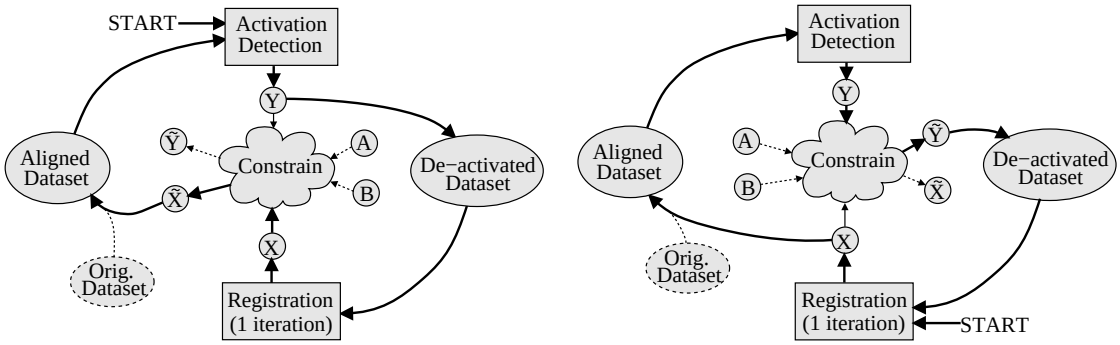


Figure 3.1: Constrained paired iteration flowcharts. In (a), activation detection is performed first, while in (b), registration is performed first. Both methods converge to the same solution.

SRA method, and thus the two iterative methods are mathematically equivalent. Recall that in the SRA method's initial, unconstrained solution, $(\mathbf{X}, \mathbf{Y})$, each row of $\mathbf{X}$ is orthogonal to each row of $\mathbf{B}$. This allows $\mathbf{X}$ and $\mathbf{Y}$ to be decoupled and solved for independently. The important thing to note is that the $\mathbf{Y}$-solution that the SRA method returns is the same as it would have been if the activation detection method were run in isolation, before registration. To demonstrate this fact, consider the activation detection problem, and the derivation

$$\begin{aligned} \min_{\mathbf{Y}} \|\mathbf{YB} - \mathbf{C}\| &= \min_{\mathbf{Y}} \left\| \mathbf{Y}[\mathbf{R}' \; 0] \begin{bmatrix} \mathbf{Q}_1' \\ \mathbf{Q}_2' \end{bmatrix} - \mathbf{C} \right\| && (3.26) \\ &= \min_{\mathbf{Y}} \|\mathbf{Y}[\mathbf{R}' \; 0] - \mathbf{C}[\mathbf{Q}_1 \; \mathbf{Q}_2]\| && (3.27) \\ &= \min_{\mathbf{Y}} \|\mathbf{Y}[\mathbf{R}' \; 0] - \mathbf{C}[\mathbf{Q}_1 \; 0]\| + \|\mathbf{C}[0 \; \mathbf{Q}_2]\| \; . && (3.28) \end{aligned}$$

The second term in (3.22) has the same $\mathbf{Y}$-solution as (3.28). Thus, simply performing the activation detection on its own yields the same activation map as the unconstrained solution in the SRA method. Hence, for any given iteration, these two methods yield the same unconstrained solution for $\mathbf{Y}$.

Removing the estimated activation from the dataset is done by subtracting the term $\mathbf{YB}$, or equivalently $\mathbf{CB}^{\dagger}\mathbf{B}$, from $\mathbf{F}$ (recall that $\mathbf{B}^{\dagger}$ is defined as $\mathbf{B}'(\mathbf{BB}')^{-1}$). Hence, $\mathbf{C}$

(which is equal to $\mathbf{F} - \mathbf{G}$) is replaced by $\mathbf{C} - \mathbf{C}\mathbf{B}^{\dagger}\mathbf{B}$ in (3.2), and we get

$$\min_{\mathbf{X}} \|\mathbf{A}\mathbf{X} - (\mathbf{C} - \mathbf{C}\mathbf{B}^{\dagger}\mathbf{B})\| \tag{3.29}$$

$$= \min_{\mathbf{X}} \|\mathbf{A}\mathbf{X} - \mathbf{C}(\mathbf{I} - \mathbf{P}_{\mathbf{B}})\| \ , \tag{3.30}$$

where $\mathbf{P}_{\mathbf{B}}$ is $\mathbf{B}^{\dagger}\mathbf{B}$, the projection operator that projects the rows of $\mathbf{C}$ onto the row-space of $\mathbf{B}$. Thus, $\mathbf{C}(\mathbf{I} - \mathbf{P}_{\mathbf{B}})$ is the projection of the rows of $\mathbf{C}$ onto the space orthogonal to the row-space of $\mathbf{B}$. The least-squares solution for $\mathbf{X}$ can be found analytically, and is

$$\mathbf{X} = \mathbf{A}^{\dagger}\mathbf{C}(\mathbf{I} - \mathbf{P}_{\mathbf{B}}) \ . \tag{3.31}$$

Recall that the rows of $\mathbf{C}(\mathbf{I} - \mathbf{P}_{\mathbf{B}})$ are orthogonal to the row-space of $\mathbf{B}$. Left-multiplying by $\mathbf{A}^{\dagger}$ simply creates rows that are linear combinations of the rows in $\mathbf{C}(\mathbf{I} - \mathbf{P}_{\mathbf{B}})$, and are thus still orthogonal to the row-space of $\mathbf{B}$. Therefore, just as in the SRA method, the CPI-A method produces an unconstrained $\mathbf{X}$ with rows that are orthogonal to all the rows in $\mathbf{B}$.

What happens if we implement the paired iteration algorithm so that registration is done first instead of activation detection, as shown in Fig. 3.1(b)? In section 3.3, it was noted that the SRA algorithm can also be implemented so that the QR decomposition of $\mathbf{A}$ is used instead of that of $\mathbf{B}$. While this option greatly increases the memory requirements of the method, the solution is mathematically equivalent. The paired iteration method in which registration is performed first (CPI-R) is analogous to the SRA method using the QR decomposition of $\mathbf{A}$.

3.6 Undetermined Baseline Volume

There is a danger to choosing a particular frame as the baseline volume, $\mathbf{g}$. If, for example, the chosen volume happens to be somewhat darker than the others, then the model will not be able to accurately represent the dataset, forcing all the frames except for the first to be treated as outliers. We can instead allow the model to calculate its own optimal baseline volume by reformulating the problem in (3.4) so that the matrix $\mathbf{Y}$ is augmented with the column-vector $\mathbf{g}$, as follows:

$$\mathbf{F} = \mathbf{G} + \mathbf{A}\mathbf{X} + \mathbf{Y}\mathbf{B} \tag{3.32}$$

$$= \mathbf{g}\,[1\ 1\ \ldots\ 1] + \mathbf{A}\mathbf{X} + \mathbf{Y}\mathbf{B} \tag{3.33}$$

$$= \mathbf{AX} + \begin{bmatrix} \mathbf{Y} & \mathbf{g} \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ 1 \end{bmatrix}, \tag{3.34}$$

where $\mathbf{Y}$ and $\mathbf{B}$ are as before, and the 1 refers to a row (time series) of ones.

Thus, by simply attaching a row of ones to $\mathbf{B}$, and a corresponding column to $\mathbf{Y}$, the optimal baseline volume can be calculated using equation (3.24) in the same way that it is used to find the activation maps. Note that in this new formulation, the matrix $\mathbf{C}$ represents $\mathbf{F}$ alone, rather than $\mathbf{F} - \mathbf{G}$ as before.

However, adding a row of ones to $\mathbf{B}$ increases the dimension of our solution space, adding a column to the matrix α in (3.11). That is, the row of ones in $\mathbf{B}$ allows every row of $\mathbf{X}$ to contain a different constant offset. We cannot use (3.25) to constrain this new freedom because the corresponding column of $\mathbf{Y}$ is not an activation map and does not have the sparsity characteristic that the constraint in (3.25) is designed to favour. That is, we do not expect the baseline volume to be sparse.

Instead, notice that we can arbitrarily choose one volume to be stationary since all the motion parameters are relative to each other. Recall that the previous general solution for the motion parameters is $\mathbf{X} + \alpha\mathbf{B}$. If the baseline volume is undetermined, the general solution can instead be expressed as

$$\mathbf{X} + [\alpha\ \beta] \begin{bmatrix} \mathbf{B} \\ 1 \end{bmatrix} \tag{3.35}$$

where β represents the new column of arbitrary constants. We simply set β to the negative of the stationary volume's motion parameters. This choice causes those values to be subtracted from the motion parameters of all the volumes, setting the stationary volume's motion to zero, and moving all the other volumes accordingly. This approach is valid only because we assumed the motion increments are small, allowing us to treat the rigid-body transformation as a linear operator.

Once the desired α and β are established, the constrained solution to the combined problem is unique. In our implementation, we let the model calculate the optimal baseline volume at each iteration, and used the Nelder/Mead simplex method [71] (also known as the Downhill simplex method) to find the columns of α.

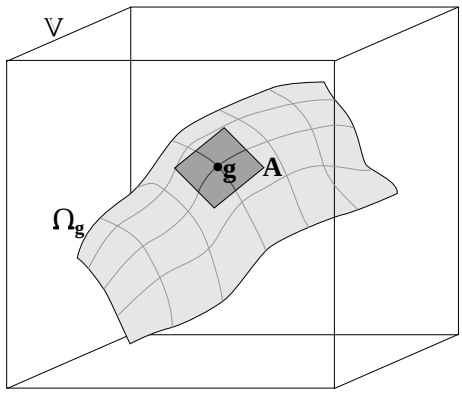


Figure 3.2: The motion manifold of $\mathbf{g}$ is $\Omega_{\mathbf{g}}$. Also shown is the tangent hyperplane, $\mathbf{A}$. In this scaled-down view, the underlying space of volumes, $\mathbb{V}$, is only 3D, and the motion manifold is 2D. This depiction is intended to give an intuitive notion of the relationship between $\mathbb{V}$ and $\Omega_{\mathbf{g}}$. In the actual fMRI dataset, the space of volumes, $\mathbb{V}$, is m-dimensional and the motion manifold is 6D.

3.7 Geometrical Interpretation

In this section, we enhance our understanding of the simultaneous method using geometrical arguments.

Recall that volumes have been represented by column-vectors. We now extend this idea slightly, and define the vector-space $\mathbb{V}$ as

$$\mathbb{V} = \{\mathbf{g} \in \mathbb{R}^m\} \tag{3.36}$$

where m is the number of voxels in a volume. Thus, every conceivable volume is a vector in $\mathbb{V}$. Now, for any given volume $\mathbf{g}$, consider the set of volumes that result from applying all possible rigid-body motion transformations to $\mathbf{g}$,

$$\Omega_{\mathbf{g}} = \left\{\mathbf{h} \in \mathbb{V} \mid \mathbf{h} = T(\mathbf{g}, \mathbf{x}),\ \mathbf{x} \in \mathbb{R}^6\right\} \ . \tag{3.37}$$

The set $\Omega_{\mathbf{g}}$ is a six-dimensional (6D) manifold, which we will call the *motion manifold* of $\mathbf{g}$. To gain an intuition for what this manifold is, consider a lower-dimensional case in which $\mathbb{V}$ is 3D, with only two motion parameters. Then, the motion manifold will be a 2D surface in 3D. Figure 3.2 depicts this simplified view of the relationship between $\mathbb{V}$, $\mathbf{g}$, $\Omega_{\mathbf{g}}$, and the hyperplane $\mathbf{A}$ which is the linearization of $T(\mathbf{g}, \mathbf{x})$.

In this context, we can begin to visualize the process of registering $\mathbf{f}$ to $\mathbf{g}$. Suppose that

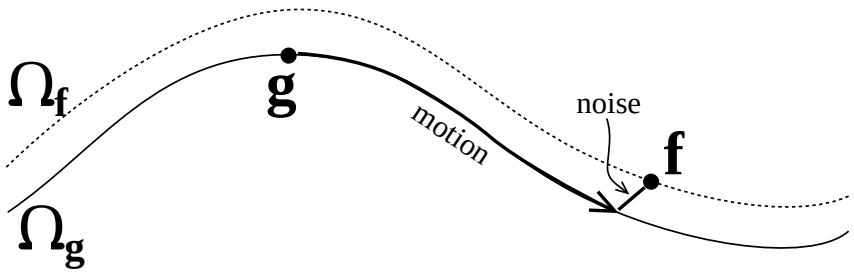


Figure 3.3: Simplified view of the motion manifold of $\mathbf{g}$. Adding noise to a moved version of $\mathbf{g}$ gives the volume $\mathbf{f}$. The motion manifold of $\mathbf{f}$ is similar to that of $\mathbf{g}$.

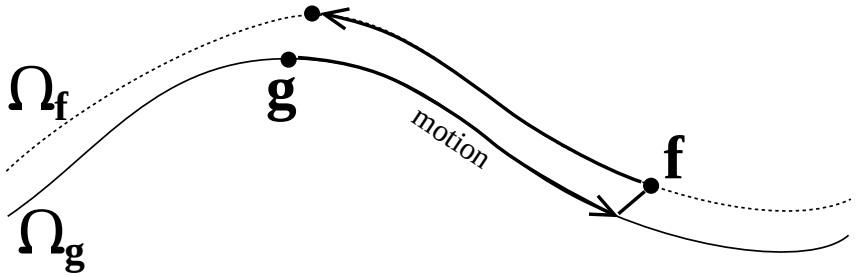


Figure 3.4: If the inverse of the true motion is applied to $\mathbf{f}$, then we attain the true aligned solution.

$\mathbf{f}$ is a moved version of $\mathbf{g}$ with some added noise

$$\mathbf{f} = T(\mathbf{g}, \mathbf{x}) + \varepsilon \ , \tag{3.38}$$

where $\varepsilon \in \mathbb{V}$ is the noise. Then, the volume $\mathbf{f}$ lies close to $\mathbf{g}$'s motion manifold, $\Omega_{\mathbf{g}}$. For the remainder of this section, we will further reduce the dimension of the visualization so that $\mathbb{V}$ is 2D, and the motion manifolds are 1D. This does not change the concepts inherent in the problem, but makes the diagrams much less cluttered. Figure 3.3 shows the the relationship between $\mathbf{g}$ and $\mathbf{f}$.

If we know the true motion that was applied to $\mathbf{g}$ to attain $\mathbf{f}$, then we can apply the inverse of that motion to $\mathbf{f}$ to get the perfectly-aligned volume $T^{-1}(\mathbf{f}, \mathbf{x})$, which we will denote $\mathbf{f}^*$. Figure 3.4 shows this re-alignment transformation. Notice that $\mathbf{g}$ and $\mathbf{f}^*$ still differ by noise.

Recall from equation (2.55) (page 47) that our fixed-point iterative registration method

can be stated as

$$\begin{cases} \mathbf{f}^{(k)} = T^{-1}\left(\mathbf{f}^{(0)}, \Sigma\mathbf{x}^{(k-1)}\right) \\ \mathbf{x}^{(k)} = \mathbf{A}_{\mathbf{g}}^{\dagger}\left(\mathbf{f}^{(k)} - \mathbf{g}\right) \end{cases}, \tag{3.39}$$

where $\Sigma\mathbf{x}^{(k-1)}$ is the sum of all previous motion increments, and $\mathbf{A}_{\mathbf{g}}$ is $\nabla T(\mathbf{g}, 0)$ and comes from the linearization of $T(\mathbf{g}, \mathbf{x})$. The first line of (3.39) is the resampling of $\mathbf{f}^{(0)}$ using the best-guess set of motion parameters, $\Sigma\mathbf{x}^{(k-1)}$. The second line is the calculation of a new motion increment using the difference between $\mathbf{f}^{(k)}$ and $\mathbf{g}$, and essentially performs an orthogonal projection of $(\mathbf{f}^{(k)} - \mathbf{g})$ onto the sub-space of $\mathbb{V}$ spanned by the columns of $\mathbf{A}_{\mathbf{g}}$. However, the fact that the linearization is anchored at $\mathbf{g}$ suggests that $\mathbf{g}$ should be moved into alignment with $\mathbf{f}^{(0)}$, rather than $\mathbf{f}^{(0)}$ moved into alignment with $\mathbf{g}$. The first two iterations of this method are illustrated in Fig. 3.5.

An alternative formulation is to use the linearization of $T^{-1}(\mathbf{f}^{(k)}, \mathbf{x})$ instead. That means that the linearization has to be re-calculated at each iteration, since $\mathbf{f}^{(k)}$ changes. Let $\mathbf{A}_{\mathbf{f}^{(k)}}$ be $\nabla T^{-1}(\mathbf{f}^{(k)}, 0)$, the 6D hyperplane tangent to $\Omega_{\mathbf{f}}$ at $\mathbf{f}^{(k)}$. The fixed-point iterative technique is then

$$\begin{cases} \mathbf{f}^{(k)} = T^{-1}\left(\mathbf{f}^{(0)}, \Sigma\mathbf{x}^{(k-1)}\right) \\ \mathbf{x}^{(k)} = \mathbf{A}_{\mathbf{f}^{(k)}}^{\dagger}\left(\mathbf{f}^{(k)} - \mathbf{g}\right) \end{cases}. \tag{3.40}$$

Two iterations of this method are illustrated in Fig. 3.6.

The two methods in (3.39) and (3.40) are not quite equivalent. Upon convergence, the motion increments go to zero. In (3.39), this means that $(\mathbf{f}^{(k)} - \mathbf{g})$ is in the null-space of $\mathbf{A}_{\mathbf{g}}$. However, convergence of (3.40) means that $(\mathbf{f}^{(k)} - \mathbf{g})$ is in the null space of $\mathbf{A}_{\mathbf{f}^{(k)}}$. If $\mathbf{A}_{\mathbf{g}}$ and $\mathbf{A}_{\mathbf{f}^{(k)}}$ are the same, then these two methods yield the same solution. However, while $\mathbf{A}_{\mathbf{g}}$ and $\mathbf{A}_{\mathbf{f}^{(k)}}$ are likely very similar, they probably are not identical. It is worth noting that neither method is likely to give perfect re-alignment, since the noise ε is unlikely to be orthogonal to either $\mathbf{A}_{\mathbf{g}}$ or $\mathbf{A}_{\mathbf{f}^{(k)}}$. The difference between these three solutions is demonstrated in Fig. 3.7.

The fact that the least-squares solutions differ from the true solution is not surprising. The noise volume, ε, is most likely not in the space orthogonal to $\mathbf{A}_{\mathbf{g}}$ or $\mathbf{A}_{\mathbf{f}^{(k)}}$. Hence, the noise will cause a small noise bias in the motion estimates. Since each volume has its own noise, the noise-induced error in registration will vary from volume to volume. We conducted some numerical studies to investigate the magnitude of these noise errors. Using a simulated volume, as described in section 4.1, Gaussian noise was added, followed by smoothing using a Gaussian kernel with a full-width at half-maximum of 5 mm. It was found that the bias due to noise was an order of magnitude larger than the difference between the two solutions

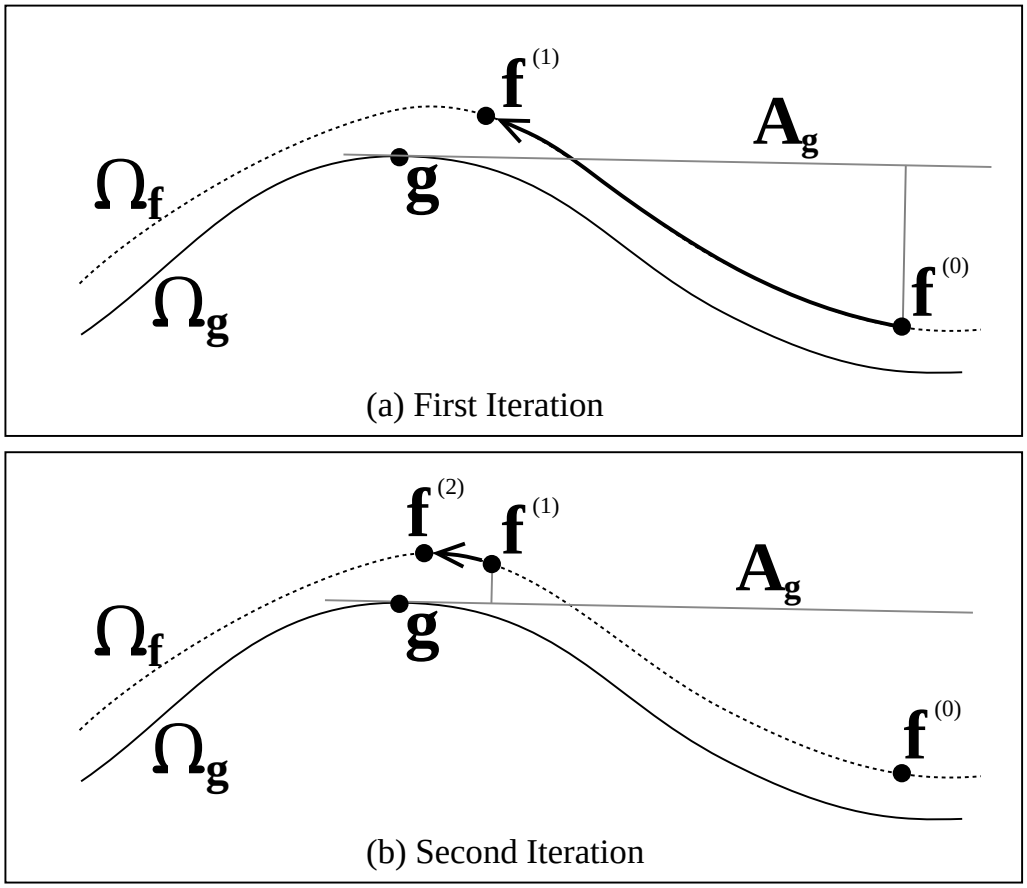


Figure 3.5: Two iterations of the fixed-point iterative registration method using $\mathbf{A_g}$. For each iteration, the motion increment is determined using an orthogonal projection of $\mathbf{f}^{(k)}$ onto $\mathbf{g}$'s tangent hyperplane, $\mathbf{A_g}$. Notice that even though $\mathbf{f}$ is being moved, the linearization of the rigid-body transformation is evaluated at $\mathbf{g}$, and $\mathbf{A_g}$ is used in each iteration. The orthogonal projection of $\mathbf{f}^{(k)}$ onto $\mathbf{A_g}$ determines $\mathbf{x}^{(k)}$, which in turn determines how far $\mathbf{f}^{(k)}$ moves along $\Omega_\mathbf{f}$.

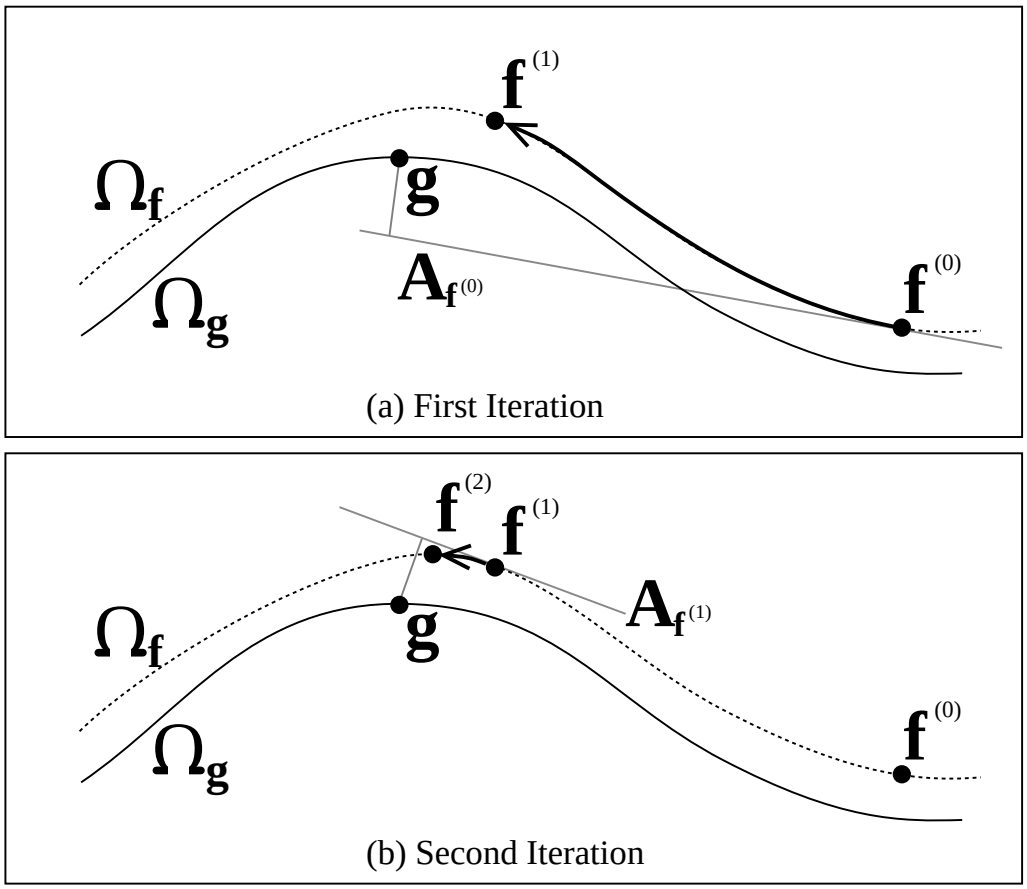


Figure 3.6: Two iterations of the fixed-point iterative registration method using $\mathbf{A}_{\mathbf{f}^{(k)}}$. At each iteration (call it k), a new linearization of the rigid-body transformation is established and anchored at $\mathbf{f}^{(k)}$, and the tangent hyperplane $\mathbf{A}_{\mathbf{f}^{(k)}}$ is re-calculated. The orthogonal projection of $\mathbf{g}$ onto $\mathbf{A}_{\mathbf{f}^{(k)}}$ determines $\mathbf{x}^{(k)}$, which in turn determines how far $\mathbf{f}^{(k)}$ moves along $\Omega_{\mathbf{f}}$.

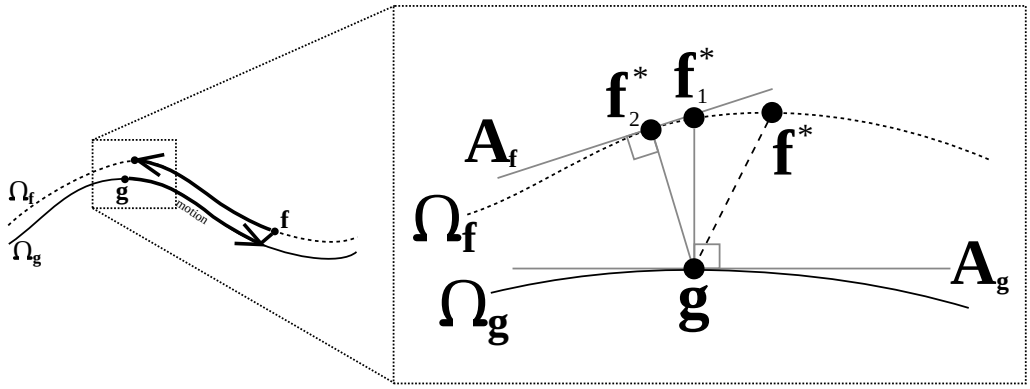


Figure 3.7: If the inverse of the true motion is applied to $\mathbf{f}$, then we attain the perfectly-aligned volume (plus noise), $\mathbf{f}^*$. If the linearization at $\mathbf{g}$ is used, the solution converged to is $\mathbf{f}_1^*$. Using the linearization at $\mathbf{f}^{(k)}$ (updating at each iteration) yields the solution $\mathbf{f}_2^*$. Notice that $\mathbf{f}_2^*$ is in fact the closest volume on $\Omega_\mathbf{f}$ to $\mathbf{g}$, and that neither $\mathbf{f}_1^*$ or $\mathbf{f}_2^*$ are likely to be the perfectly-aligned solution.

$\mathbf{f}_1^*$ and $\mathbf{f}_2^*$. For this reason, for the remainder of this thesis, we consider the two solutions to be effectively equivalent, and always use the linearization of the rigid-body transformation at $\mathbf{g}$, and take $\mathbf{A}$ to equal $\mathbf{A_g}$. Furthermore, the bias caused by the noise was an order of magnitude smaller than the activation bias outlined in section 1.4.

Similar geometrical analysis can be done for the activation detection least-squares problem. However, since the activation detection problem really *is* linear, no linearization is required, and the least-squares solution is attained without iteration. Noise still induces a small bias in the solution of the activation detection problem, causing the least-squares solution to be slightly different from the true solution. The geometry of these solutions is illustrated in Fig. 3.8.

The above geometrical analysis operates on the vector-space of volumes, $\mathbb{V}$. In the combined problem, we seek a solution to both the registration and activation problems simultaneously. In doing so, we define the dataset matrix as a matrix that holds one volume in each column. We now extend this analysis to operate on the vector-space of datasets, and define *dataset space* as

$$\mathbb{D} = \left\{ \mathbf{G} \in \mathbb{R}^{m \times n} \right\} \ . \tag{3.41}$$

Likewise, we define the motion manifold for a dataset $\mathbf{G} \in \mathbb{D}$ as

$$\Omega_\mathbf{G} = \left\{ \mathbf{H} \in \mathbb{D} \mid \mathbf{H} = T(\mathbf{G}, \mathbf{X}),\ \mathbf{X} \in \mathbb{R}^{6 \times n} \right\} \ . \tag{3.42}$$

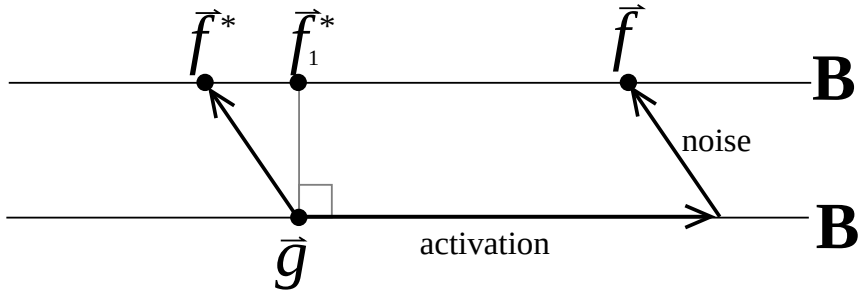


Figure 3.8: If the true activation is removed from $\vec{f}$, then we attain the perfectly-de-actived volume $\vec{f}^*$. However, the solution to the least-squares activation detection problem, $\vec{f}^*_1$, will likely be slightly different than $\vec{f}^*$ because the noise is likely not orthogonal to $\mathbf{B}$. The noise bias is exaggerated in this illustration.

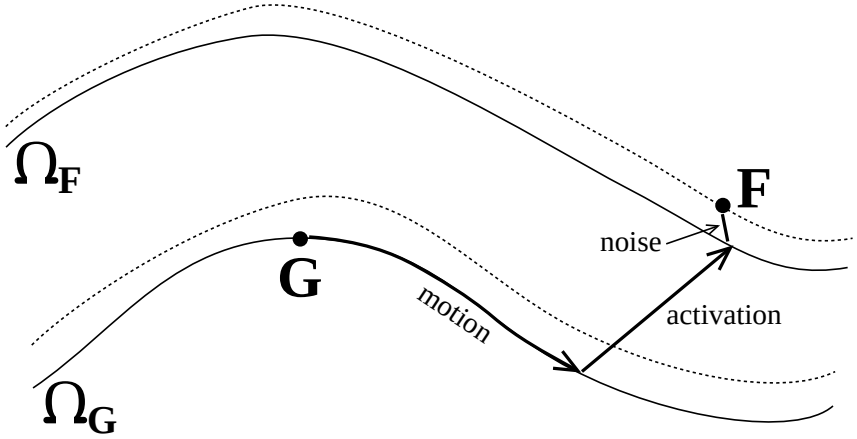


Figure 3.9: Simplified view of the motion manifold of $\mathbf{G}$. The dataset $\mathbf{F}$ is the result of adding motion, activation and noise to the baseline dataset $\mathbf{G}$.

The datasets $\mathbf{G}$ and $\mathbf{F}$ can differ by motion, activation, and noise, as shown in Fig. 3.9. Using these geometrical illustrations, we can better understand the nature of activation bias. Just like noise, the displacement in $\mathbb{D}$ caused by activation is likely not orthogonal to $\mathbf{A}$, and therefore creates a bias in the motion estimates. Figure 3.10 demonstrates the activation bias. The perfectly-aligned dataset is the dataset in $\Omega_\mathbf{F}$ that equals $\mathbf{G} + \mathbf{YB}$ (plus some noise). However, if we ignore the presence of the activation and perform registration alone on $\mathbf{F}$, we will arrive at the dataset $\mathbf{F}^*_\mathrm{R}$. The activation bias, as shown in Fig. 3.10, is the difference in the motion parameters between the perfectly-aligned dataset and $\mathbf{F}^*_\mathrm{R}$.

Solving the combined problem using the SRA method is also iterative. At each iteration,

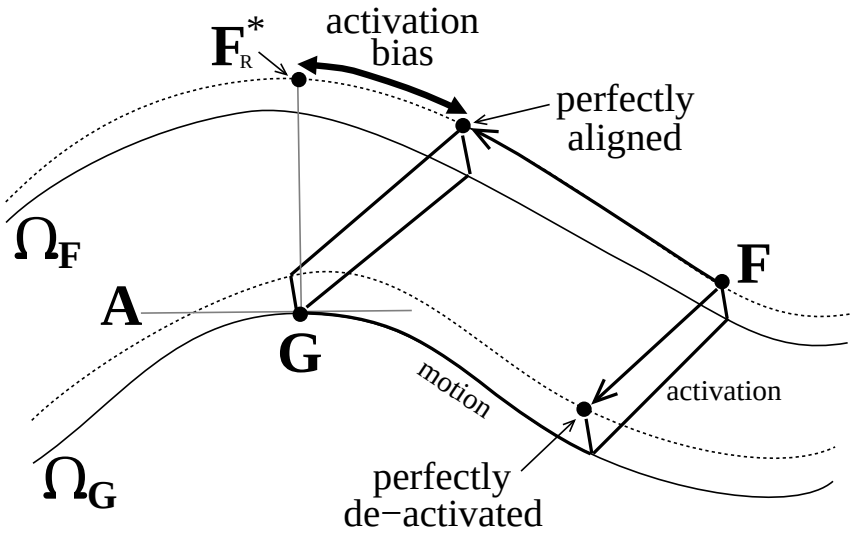


Figure 3.10: The true registration solution moves $\mathbf{F}$ so that it equals $\mathbf{G} + \mathbf{YB}$ + noise. However, if the activation is ignored, the registration algorithm will yield the solution $\mathbf{F}_\mathrm{R}^*$.

the method generates an initial (unconstrained) solution for both motion ($\mathbf{X}$) and activation ($\mathbf{Y}$). As discussed in section 3.2, this solution is probably not the one we want. The solution to the combined problem is not unique, and so far we have not enforced any preference for one solution in the space of solutions over another. For the sake of demonstration, suppose we used this initial solution $(\mathbf{X}, \mathbf{Y})$ to re-align our dataset and remove the activation. We would get another dataset, $\mathbf{H}$, that contains both motion and activation artifacts. However, since all the solutions in our solution space yield equivalent sum-of-squares cost, the set of corresponding datasets forms a hypersphere with $\mathbf{G}$ at its centre. Thus, no matter what solution we choose from the solution space, $\mathbf{H}$ remains the same distance from $\mathbf{G}$ (this is true because the Frobenius norm is equivalent to Euclidean distance, in our application).

Two iterations of the SRA method are illustrated in Fig. 3.11, showing the initial motion correction, the initial activation removal, as well as the adjustment arrived at by enforcing the additional constraint in (3.25). Notice that the constraint simply moves our dataset on an arch, remaining equidistant from $\mathbf{G}$.

The point of this geometrical analysis is to both offer some intuitive, geometrical basis behind these least-squares solutions, and to demonstrate the close relationship between the SRA method and the standard least-squares registration method. The SRA algorithm, just like the standard, uses a linear estimate of the motion to bring the dataset closer (in

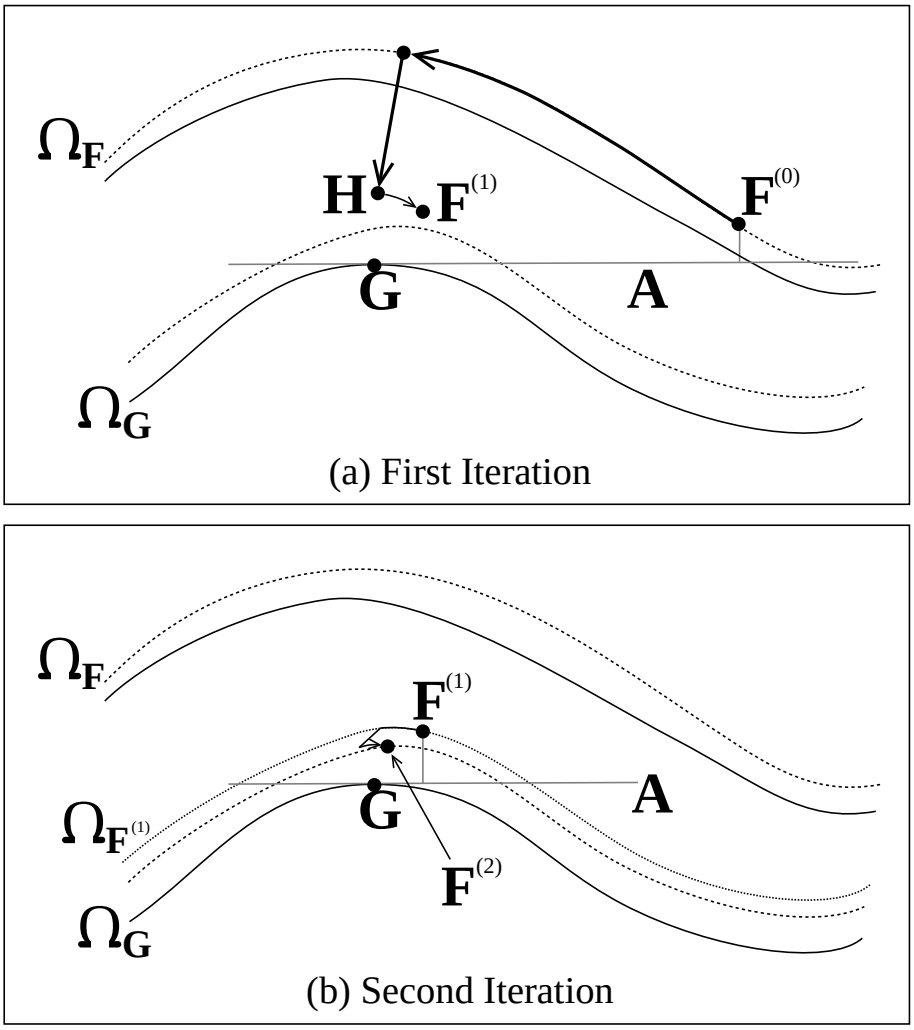


Figure 3.11: Two iterations of the SRA method. Each iteration consists of an initial estimate of motion and activation. If those un-adjusted estimates were used, the dataset $\mathbf{H}$ would result. However, the fact that the least-squares solution is not unique means that any dataset that is the same distance from $\mathbf{G}$ is an equivalent solution. Applying the additional constraint (see equation (3.25)) moves us in a circular arch to $\mathbf{F}^{(1)}$. Carrying out the process for another iteration takes us to $\mathbf{F}^{(2)}$.

Frobenius norm, or Euclidean distance) to the baseline dataset, **G**. Thus, we can expect the asymptotic behaviour of the two iterative methods to be similar. That is, the convergence behaviour of the SRA method is inherited from the least-squares registration process.

Chapter 4

Testing the Simultaneous Methods

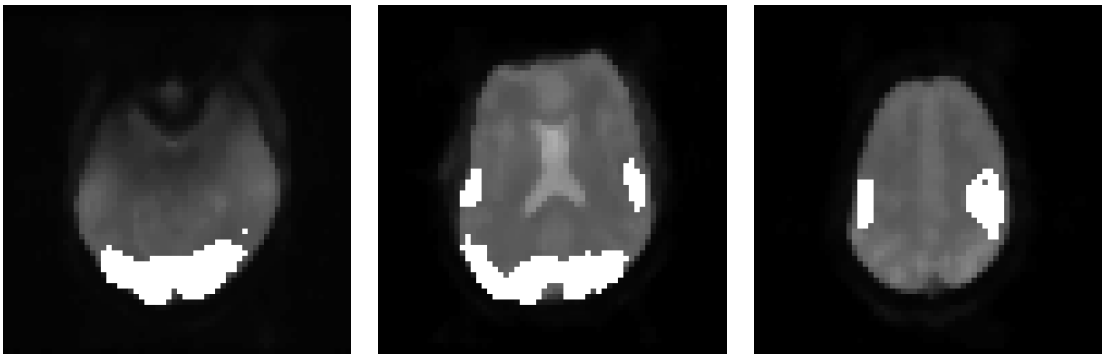
4.1 Simulated Datasets

To establish the accuracy of the analysis methods outlined in this thesis, it is essential to have datasets with known motion and activation. Motion errors on the order of 0.05 mm (1/75 of a voxel) and 0.05° can confound the activation detection [30]. However, it is unrealistic to expect even a cooperative subject to move less than that. Moreover, the true activation pattern is not known for a real human subject. Determining the accuracy of the SRA method on *in vivo* data without known motion or activation is not possible.

Most of the testing of analysis methods here is done on simulated datasets. These datasets were created using an initial $64 \times 64 \times 30$ EPI volume with in-plane resolution of 3.75 mm and a slice thickness of 4 mm. The activation template, shown overlaid on the original volume in Fig. 4.1, was manually drawn covering a large portion of the occipital region, as well as some regions in the parietal lobes. The entire activation template covers about 13% of the total brain volume. A stimulus function was created using a "box-car"-like function, as shown in Fig. 4.2. The stimulus function's maximum value is 0.05, indicating that a voxel's intensity will change by as much as 5% due to the artificial BOLD signal.

Both motion-free and motion-corrupted simulated datasets were generated using the following sequence:

1. Duplicate the original volume 40 times.
2. Add the appropriate amount of activation to the volumes. Activation is added to voxels under the activation template by multiplying their intensities by $(1 + b)$, where



(a) Slice 10 (b) Slice 16 (c) Slice 23

Figure 4.1: Sample slices of the activation template used to create the simulated datasets. The template is overlaid on a darkened version of the original EPI volume.

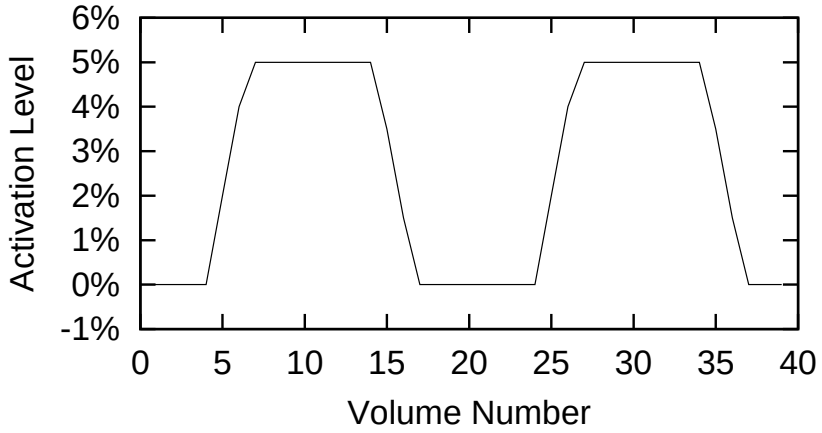


Figure 4.2: Stimulus function for simulated datasets. The elevated portions are not exactly symmetrical, although symmetry should not change any of the results.

b is the relative signal change due to BOLD contrast, specified by the stimulus function (see Fig. 4.2).

3. Apply rigid-body transformations (motion-corrupted datasets only). For this operation alone, the volumes are super-sampled by a factor of two in each dimension to reduce resampling artifacts. The transformations are applied in the frequency domain using Fourier interpolation, with rotations implemented as a composition of four shears [21, 23]. Fourier interpolation is generally considered to be among the most accurate interpolation techniques [40], especially for small rotations [29].

4. Add Gaussian-distributed, multiplicative noise ($\mu = 0$, $\sigma = 2.5\%$).

5. Apply spatial Gaussian smoothing to each volume (the Gaussian kernel has a full-width at half-maximum of 5 mm).

To isolate the strengths and weaknesses of the various algorithms discussed here, datasets containing a variety of different mixtures of motion and activation were generated. Some datasets contained random motion. These random motion profiles were simply random walks. That is, starting from zero, Gaussian-distributed random numbers were generated and accumulated in a running sum. The motion profiles are generally within the range -1 to 1 (mm or degrees). This is a typical range of motion for a cooperative patient.

Some datasets were given stimulus-correlated motion. Stimulus-correlated motion profiles were created by adding together a random motion profile and a randomly-weighted stimulus function. Thus, each motion parameter contained both random and stimulus-correlated motion.

Ten datasets were generated for each of the following four scenarios:

Scenario 1: The dataset contains activation and random motion.

Scenario 2: The dataset contains activation and true stimulus-correlated random motion.

Scenario 3: The dataset contains stimulus-correlated random motion, but no activation.

Scenario 4: The dataset contains activation, but no motion.

We did not focus on datasets without activation because activation bias is not an issue. Hence, we did not generate simulated datasets with no motion and no activation or with

random motion and no activation. Note, however, that scenario 4 datasets contain no activation, but also stimulus-correlated random motion. These datasets were included in our experiments to test each method's ability to distinguish stimulus-correlated motion from activation.

4.2 Generation of Activation Masks

The final product of an fMRI experiment is the activation map. This is usually portrayed as a binary mask, highlighting the voxels that are considered to be active. An activation mask is not quite the same thing as an activation map. An activation *map* assigns a value (or values) to each voxel, indicating the degree to which the voxel is active. These values can be integers, floating-point numbers, or even vectors. An activation *mask*, on the other hand, assigns a binary value to each voxel. The binary value is simply a yes-or-no (true-or-false) flag that states whether or not the voxel is considered to be active. This mask is usually determined by applying a thresholding operation to the activation map.

The following procedure outlines the steps taken to calculate a binary activation mask for any given dataset in our study.

The final motion estimates (determined by any of the registration methods) are used to resample the dataset before an activation map is generated. Calculation of the activation map is done using the general linear model, and solved in the least-squares sense (see section 2.2.3) yielding a linear fit coefficient for each voxel/stimulus regressor combination. The linear fit coefficient is a measure of how well a voxel's time course fits that of the stimulus regressor. The larger the coefficient, the greater the influence of that regressor on that voxel.

We also calculate the correlation coefficient of each voxel's time-course with each stimulus function. The formula for the correlation coefficient is the same as that in (2.1).

For each regressor, both the correlation coefficient and the linear fit coefficient maps are then thresholded, and their intersection is taken to be the activation mask for the corresponding stimulus.

The correlation coefficient is thresholded because it is a normalised value in the range [-1,1]. This allows easy comparison between datasets with a different number of time-series samples. Assuming Gaussian noise, the probability that a random time series of n samples

will yield a correlation coefficient outside the range $[-r, r]$ is [8]

$$P = 1 - \frac{2}{\sqrt{\pi}} \int_0^{r\sqrt{\frac{n}{2}}} e^{-t^2} dt \ . \tag{4.1}$$

Thresholding the linear fit coefficient is done because it rules out voxels that show a high correlation, but have a low intensity. For example, background voxels should not contain activation, and are assumed to be just noise. However, the sheer number of them makes it very likely that some of them will have time series that are, by chance, highly correlated to the stimulus. We wish to ignore these background voxels. As such, a threshold value of approximately 5% of the maximum fit coefficient is used. Only voxels with a linear fit coefficient greater than the threshold are considered candidates for the final activation mask. In our simulated datasets, the voxel intensities range from 0 to 10,000. Most brain voxels have intensities between 3,500 and 8,000. Based on these limits, the maximum BOLD signal in the brain is 5% of 8,000, which is 400. Five percent of that maximum BOLD signal is 20. Thus, a threshold value of 20 is used on the linear fit coefficient activation maps.

Finally, the two thresholds are applied, and only voxels whose correlation coefficient is above 0.505 (corresponding to $P = 0.0014$) and whose linear fit coefficient is above 20, are labeled active and included in the binary activation mask.

A true activation mask is also produced using the same process, but from a dataset that was neither motion corrupted nor motion compensated. The mask contains a total of 3454 active voxels. We use this derived activation mask instead of the original activation template in order to allow direct comparison with the activation masks of other simulated datasets. Because the simulated datasets are spatially smoothed, their activation gets spread to nearby voxels. This spreading effect is not present in the original activation template, so even a simulated dataset that is perfectly aligned will result in many false-positive activations.

The activation masks generated from the simulated datasets were compared to the true activation mask yielding false-positive and false-negative counts.

4.3 Form of the Additional Constraint

The form of the additional constraint in section 3.4 allows some flexibility in the model. Different choices for the cost function in the constraint lead to different solutions. Three different cost functions were studied for this thesis:

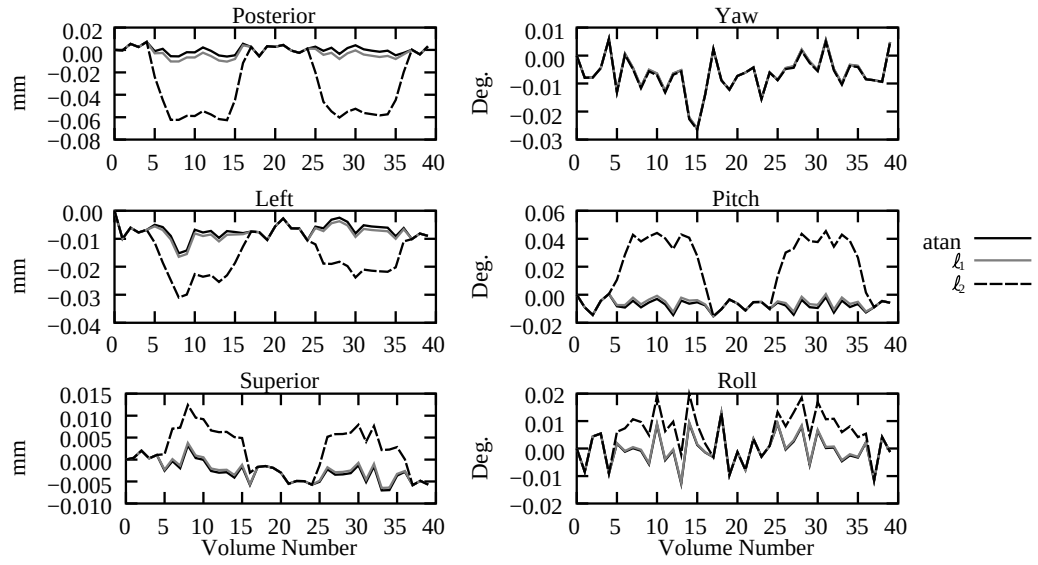


Figure 4.3: Motion parameter errors for the SRA method using three different additional constraints: arctan, ℓ_1, and ℓ_2 (see equations (4.2) (4.3) and (4.4), respectively). The plots for the ℓ_2 constraint show errors that are correlated to the stimulus function shown in Fig. 4.2. For a definition of the motion parameters, see the caption of Fig. 1.5.

arctan **cost function**

$$\min_{\alpha} \sum_{i} \arctan\left(c|\mathbf{Y}_i - [\mathbf{A}\alpha]_i|\right) \tag{4.2}$$

ℓ_1 **norm cost function**

$$\min_{\alpha} \|\mathbf{Y} - \mathbf{A}\alpha\|_1 \tag{4.3}$$

ℓ_2 **norm cost function**

$$\min_{\alpha} \|\mathbf{Y} - \mathbf{A}\alpha\|_2 \tag{4.4}$$

Figure 4.3 plots the motion parameter errors returned by the SRA algorithm when using each of the three additional constraints. The corresponding dataset contains activation, but no motion. The ℓ_2 cost function exhibits stimulus-correlated motion errors, while the other two cost functions do not.

For any single activation map, the cost function has to be minimized over the six variables in the corresponding column of α. To gain a better understanding of the behaviour of each cost function, cross-sectional plots were created by varying each of these six variables, one

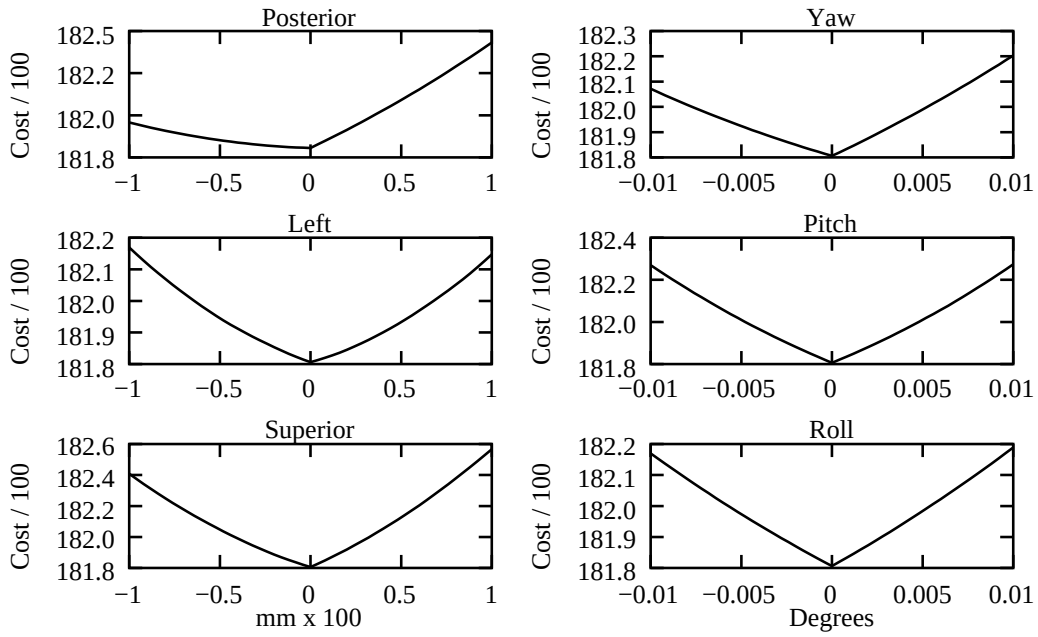


Figure 4.4: Value of the arctan cost function in (3.25) with respect to changes in the coefficients in α. The parameter c was set to 0.05. Note that the horizontal axes of the translation graphs are in units of 1/100 of a millimetre.

at a time. The true activation map, $\mathbf{Y}$, and the matrix $\mathbf{A}$ were taken from a non-motion-corrupted simulated dataset, and used to evaluate the cost functions. Since $\mathbf{Y}$ is the true activation map, the desired solution corresponds to an α of zeros. All the elements of α were set to zero except for the one being varied. Each cost plot achieves a minimum at some α value. Since the true activation map corresponds to $\alpha = \vec{0}$, the closer the minimum location is to zero, the more effective that cost function is at removing components of $\mathbf{A}$. Recall that we are assuming that our desired activation maps have the sparsity characteristic and contain mostly low intensities, with only a few relatively small clusters of high intensity (the regions of activation).

Figures 4.4 to 4.7 show these cost function cross-sections. The horizontal axes are labeled with the same units as the motion plots because both products $\mathbf{AX}$ and $\mathbf{A}\alpha$ need to have the same units. Elements of $\mathbf{X}$ have units of mm or degrees, and therefore so must the elements of α.

In Fig. 4.4, the minimum of each curve is clearly located very near zero. The only remarkable feature is the highly asymmetric quality of the posterior-anterior translation

plot. However, the minimum is still at or near zero, indicating that the arctan cost function has a preference for our desired activation map, and yields an increased cost when any component of $\mathbf{A}$ is added. Also, in most cases, the minimum is a well-defined cusp.

For any given dataset, a value for c must be specified to evaluate the arctan cost function. The trick to setting a value for c is to establish an approximate activation-map threshold between the intensity of the active regions and the intensity of the inactive regions (background noise and edge artifacts). In the activation map, voxels that are inactive contain either background noise or edge artifacts caused by components from $\mathbf{A}$. The function $\arctan\theta$ is a bounded, monotonically increasing function with about 84% of its vertical extent occurring when θ is in the range $[-4, 4]$. The remaining 16% is left for the tails, $\theta < -4$ and $\theta > 4$. Thus, choosing c so that the intensities of the inactive voxels get mapped to within the range $[-4, 4]$ shifts the focussing of the minimization off the active regions, and onto the inactive regions. Ensuring that the intensities of the active voxels stays outside this range will de-emphasize their effect on the cost. In our simulated datasets, most of the inactive voxels in the activation maps have intensities in the range $[-50, 50]$, while the active voxels have intensities mostly in the range [110, 230]. The value of 80 was chosen as the divider between the two classes. Thus, we chose our c-value to be 4/80, or 0.05. The arctan function with a c-value of 0.05 is shown in Fig. 4.5. It is worth mentioning that the c parameter is not a binary threshold, but rather a parameter defining a fuzzy boundary because the cost function varies continuously with respect to changes in c.

It should be noted that setting the parameter c in (4.2) to a very small number makes the arctan cost function equivalent to the ℓ_1 norm. The function $\arctan\theta$ is nearly linear when θ is in the range $[-0.5, 0.5]$. If c is so small that all the residuals get scaled into that range, then the cost function essentially acts like a linear, or ℓ_1, cost function. On the other hand, if c is very large, then the majority of the residuals will be scaled into the tail of the arctan function, where its value levels off asymptotically to $\pm\pi/2$. For those residuals, the cost is nearly constant, so adjusting them will have little effect on the overall cost. In such cases, the cost will depend only on the very small residuals that, even after being scaled up by a factor of c, are still in the $[-1, 1]$ range near the origin.

The cost function cross-sections for the ℓ_1 norm are shown in Fig. 4.6. The plots are similar to those of the arctan cost function, except that the ℓ_1 cost function's minimum in the posterior translation graph is not at zero, but closer to -0.01 mm (note that the horizontal axis is in units of 1/100 of a millimetre). Appropriately, the motion parameter

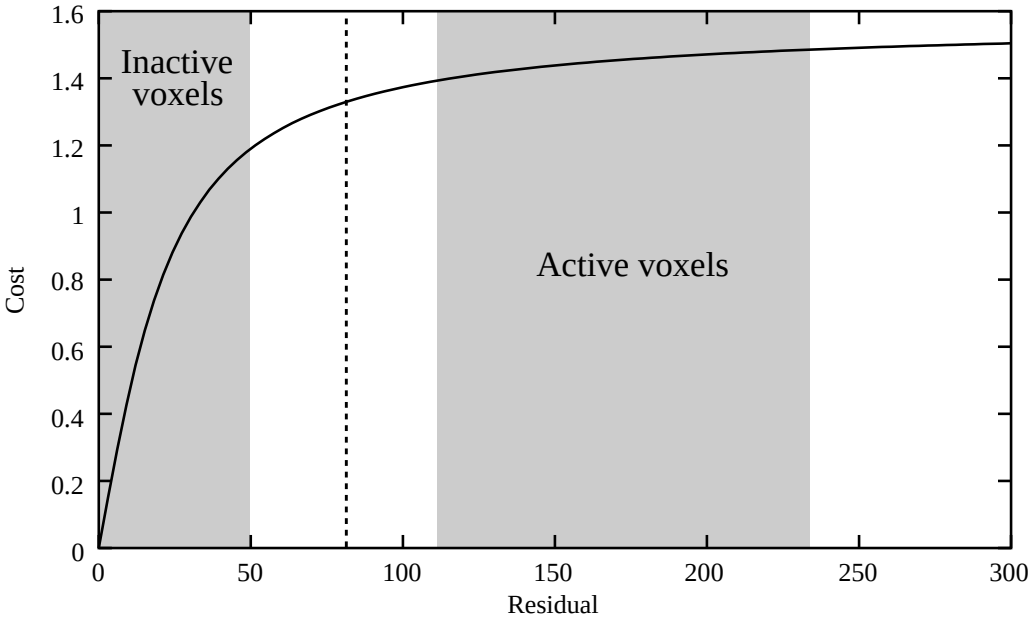


Figure 4.5: The arctan cost function with a c-value of 0.05 maps residuals in the range $[0, 80]$ into the range where the majority (84%) of the function's vertical extent occurs.

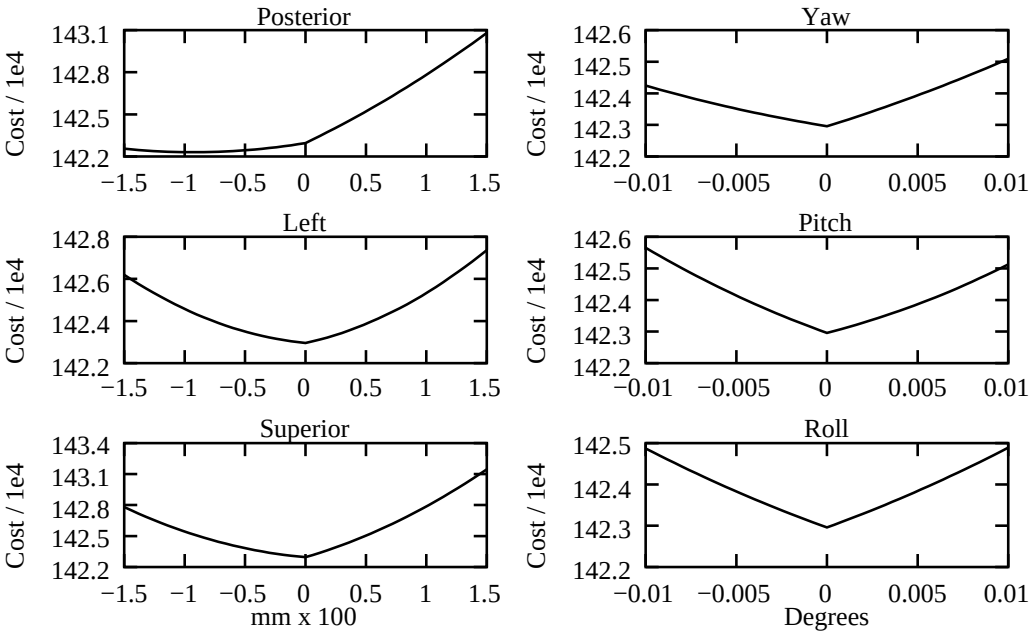


Figure 4.6: Value of the ℓ_1 cost function in (4.3) with respect to changes in the coefficients in α. Note that the horizontal axes of the translation graphs are in units of 1/100 of a millimetre.

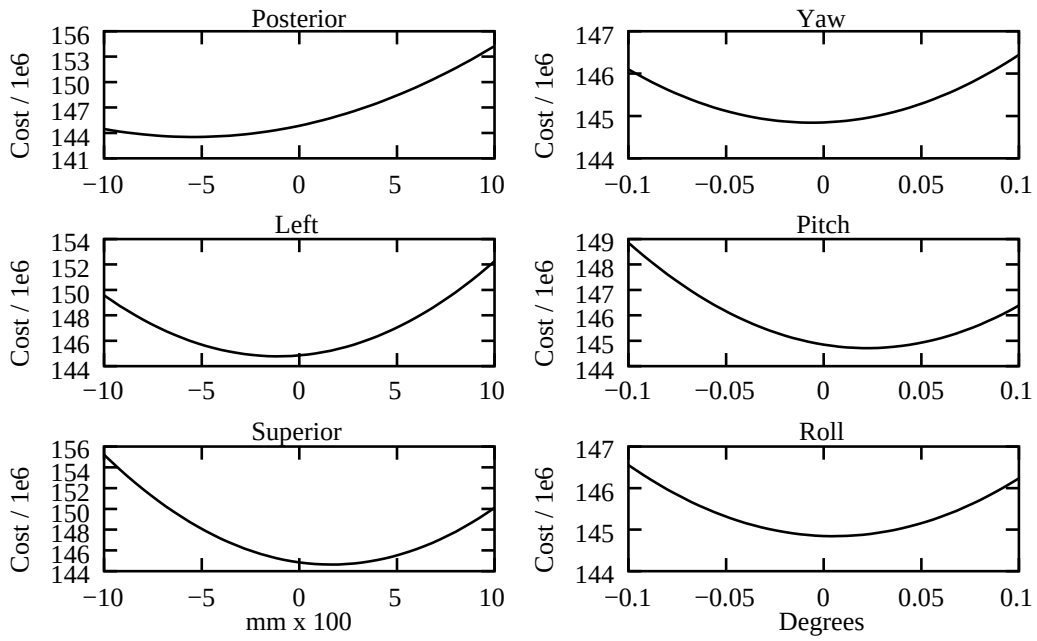
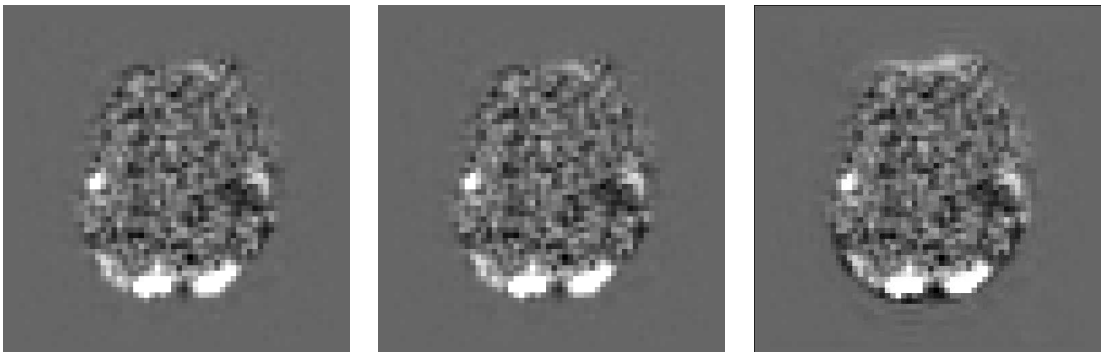


Figure 4.7: Value of the ℓ_2 cost function in (4.4) with respect to changes in the coefficients in α. Note that the horizontal axes cover ranges much larger than those in Figs. 4.4 and 4.6.

with the greatest discrepancy between the arctan and ℓ_1 cost functions in Fig. 4.3 is the posterior translation, although they differ by considerably less than 0.01 mm.

The ℓ_2 cost function cross-sections shown in Fig. 4.7 contain minima that are substantially displaced from zero. This is not surprising, given the motion errors exhibited by the ℓ_2 cost function in Fig. 4.3. In particular, the posterior translation parameter shows a minimum at -0.05 mm. This value approximates the activation-induced displacement of the posterior translation motion parameter shown in Fig. 4.3, in which the ℓ_2 cost function resulted in an erroneous posterior translation of about -0.06 mm. Also of note is the pitch angle parameter, showing a minimum at approximately 0.025°. The associated error in the motion parameter itself is about 0.04°.

Although these graphs give a rough approximation to the resulting motion parameter errors, they are meant as a tool for observing the ability of a cost function to recognize the desired solution, not to derive the motion parameters themselves. For example, even though each cross-sectional plot shows a minimum in the one-dimensional graph, the true minimum of the six-dimensional cost function is unlikely to occur at any of those exact values, but rather nearby.



(a) arctan Norm (b) ℓ_1 Norm (c) ℓ_2 Norm

Figure 4.8: Activation maps that minimize (a) the arctan cost function, (b) the ℓ_1 cost function, and (c) the ℓ_2 cost function. The gray-tone in the background corresponds to an intensity of zero, and the images have been contrast enhanced to the same range to make the relevant artifacts more apparent and directly comparable.

Figure 4.8 shows the activation maps that result from these three different additional constraints. The images for the arctan and ℓ_1 cost functions (Figs. 4.8(a) and 4.8(b)) are very similar, exhibiting bright patches at two posterior regions, as well as smaller bright patches in the left and right parietal regions. In addition to those regions of activation, the constrained activation map for the ℓ_2 cost function (Fig. 4.8(c)) also has a bright region at the anterior edge of the brain, and a dark lining along the posterior edge. These edge artifacts indicate the presence of a component from a column of $\mathbf{A}$, the partial derivative of the base volume with respect to posterior-anterior translation.

All 40 simulated datasets (10 datasets from each of the four scenarios outlined in section 4.1) were run through the SRA method three times each, once for each of the three different additional constraints. For each run, an activation mask was produced as described in section 4.2, and false-positive and false-negative activation counts were tabulated and averaged for each of the four dataset scenarios. The results are summarized in Table 4.1. The arctan and ℓ_1 constraints performed similarly on all datasets, although the arctan constraint consistently resulted in about 5% (2 standard deviations) fewer false-negatives than the ℓ_1 constraint. On datasets with activation, both the arctan and ℓ_1 constraints outperformed the ℓ_2 constraint, yielding approximately two-thirds fewer false-positive activations and one-third fewer false-negative activations.

Table 4.1: False-positive and false-negative activation counts for the three different constraints in the SRA algorithm, averaged over 10 trials for each dataset scenario. The true activation mask contains 3454 active voxels.

	False-Positives			False-Negatives		
Dataset Characteristics	arctan	ℓ_1	ℓ_2	arctan	ℓ_1	ℓ_2
activation, random motion	186.3	184.5	595.1	427.9	451.5	603.8
activation, stim-corr motion	205.6	203.3	641.7	429.0	455.9	601.2
no activation, stim-corr motion	38.4	39.4	38.2	-	-	-
activation, no motion	187.8	184.5	592.7	432.3	455.7	609.4

Since the arctan cost function shows the most accurate results in the false-positive and false-negative activation counts, the remainder of this thesis uses the arctan constraint in the SRA methods.

4.4 SRA versus Other Methods

The SRA method (using the arctan constraint) was compared to two other common registration methods: standard least-squares registration, and normalized mutual information (see page 43 in section 2.3.3). The normalized mutual information (NMI) method was chosen because it has been shown to be one of the most accurate and robust methods for registering serial MR images [46].

For this study, we used the implementation of NMI by the FMRIB Image Analysis Group at Oxford. They have a suite of programs called the FMRIB Software Library (FSL) for manipulating and analyzing fMRI datasets. For our tests, we used FMRIB's Linear Image Registration Tool (FLIRT) version 3.1 [49], with the number of bins set to 256. The program uses trilinear interpolation to resample the volumes.

Figure 4.9 shows the errors in the six motion parameters when the standard, NMI, and SRA registration methods are applied to a motion-free simulated dataset. The method labeled "Std" refers to the standard, sequential algorithm. The standard least-squares and NMI registration methods each show large stimulus-correlated errors, even though the dataset is actually motion-free. This stimulus-correlated error is not present in the estimates returned by the SRA method.

Similar results were observed for motion-corrupted simulated datasets. Figure 4.10 plots

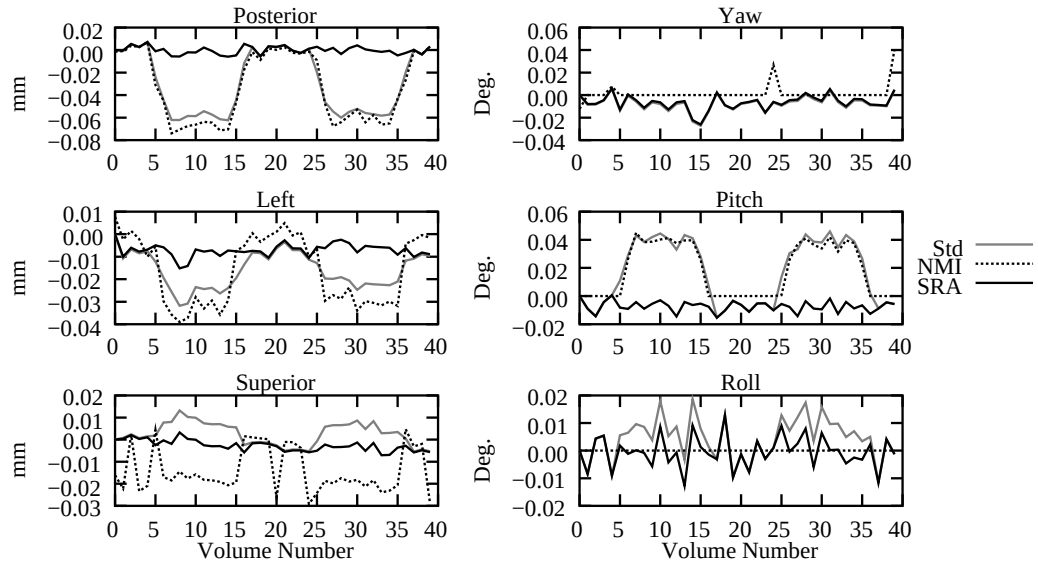


Figure 4.9: Motion parameter errors for the standard least-squares (Std), normalized mutual information (NMI), and SRA methods. The underlying dataset is motion-free.

the errors in the motion parameters for the three methods. Again, both the standard least-squares and NMI registration methods exhibit errors that are highly correlated to the stimulus, while the SRA method does not.

The activation masks corresponding to the motion plots in Fig. 4.10 are shown in Figs. 4.11 and 4.12. In the map resulting from the standard and NMI approaches, regions of false-positive activation are evident at the anterior and posterior edges of the brain. The corresponding activation mask from the SRA method looks very much like the true activation, with virtually no false-positive regions of activation.

As expected, clusters of false-activation form near contrast borders. The location of the cluster in Fig. 4.11(b) is appropriate for the motion errors exhibited by the standard method in Fig. 4.10. In the slices just superior to that shown in Fig. 4.11, the frontal lobe of the brain extends to the anterior of the head to occupy the volume adjacent to the cluster. Compensating the dataset using the motion parameters in Fig. 4.10 introduces erroneous pitch rotation and anterior-posterior translation that is synchronized to the stimulus. The resulting nodding of the head causes the frontal lobe to dip into the slice to its inferior, creating the appearance of activated voxels. Similar artifacts are observed in other parts of

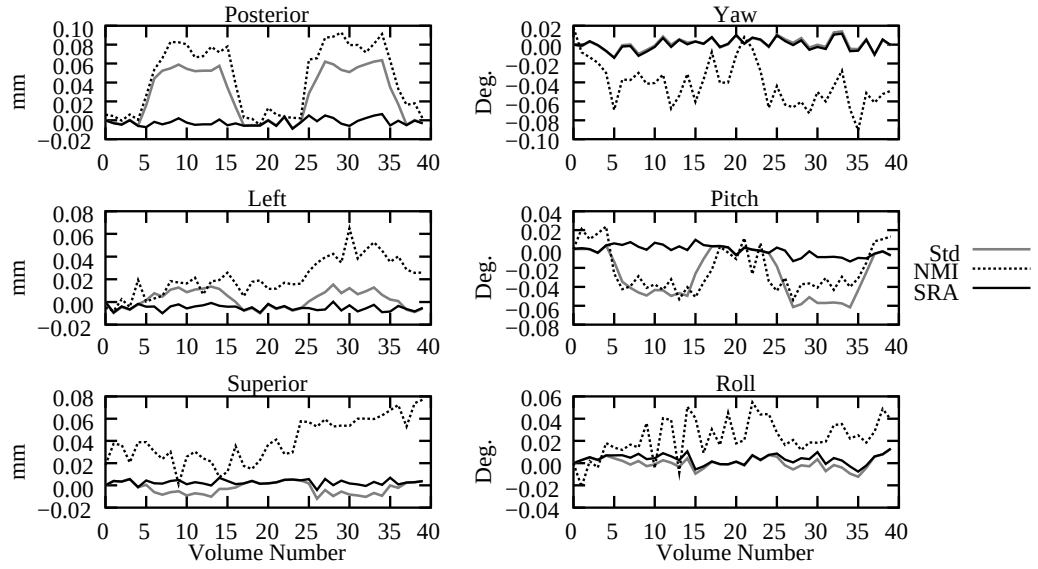


Figure 4.10: Motion parameter errors for the standard least-squares (Std), normalized mutual information (NMI), and SRA methods. The underlying dataset is motion-corrupted with random motion.

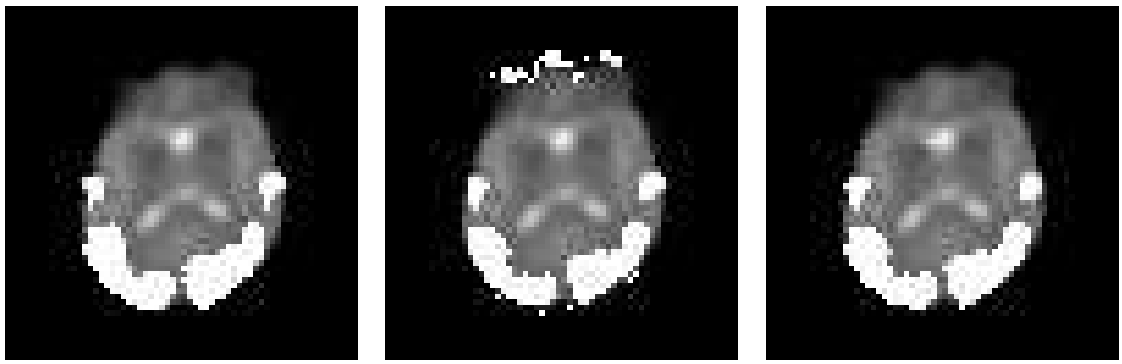
the activation map.

The performance of the three methods (standard least-squares, normalized mutual information, and SRA) is summarized in the false-positive and false-negative activation rates shown in Fig. 4.13. On datasets containing activation, the SRA method consistently resulted in substantially fewer false activations than either of the other two methods.

Results for the NMI method have been included here to lend evidence to the proposition that the least-squares cost function is not the only cost function susceptible to activation bias. However, a comprehensive comparison of the activation bias exhibited by different registration cost functions is beyond the scope of this thesis. For this reason, only least-squares-based methods will be considered from this point on.

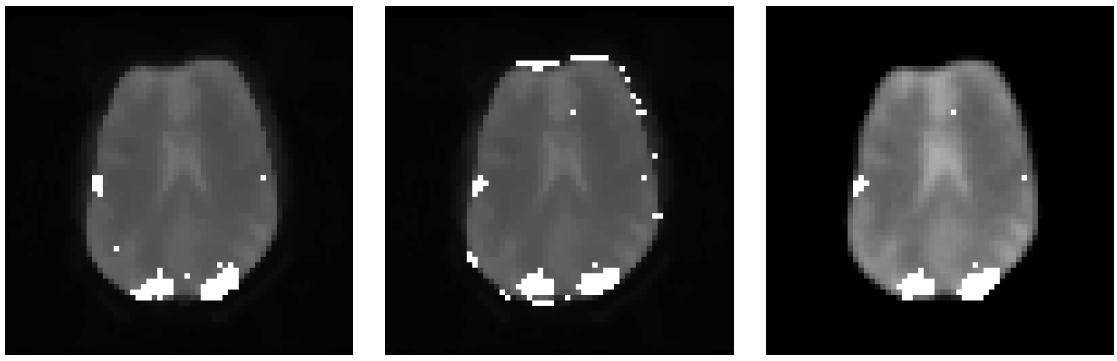
4.5 Paired Iteration

Even though the constrained paired iteration (CPI) methods detailed in section 3.5.1 have been shown to be equivalent to the SRA method, their numerical implementations can be



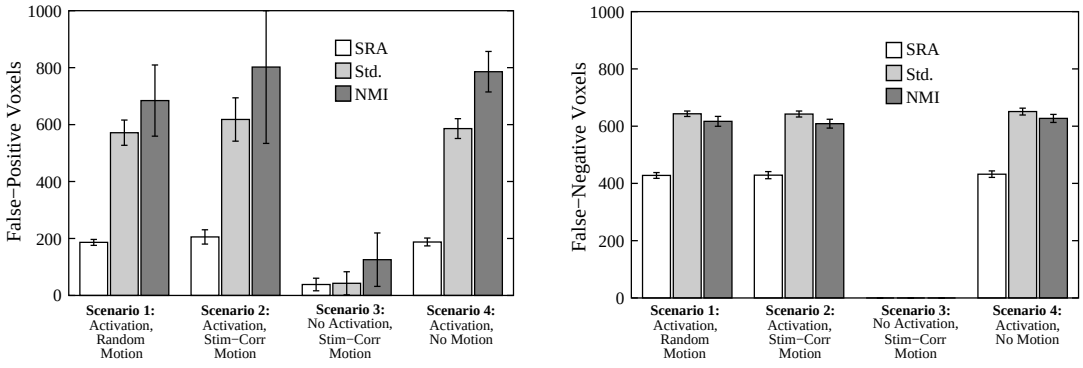
(a) True Activation (b) Standard Method (c) SRA Method

Figure 4.11: Slice 13 of the activation masks for a motion-corrupted dataset. The highlighted voxels exhibit high correlation (positive or negative) to the stimulus (significance $p = 0.0014$). The activation mask in (a) shows the actual, uncorrupted activation, while (b) shows the activation resulting from motion compensation based on the estimates from the standard method, and (c) shows the results after motion compensation based on the motion estimates returned by SRA. Similar regions of false-positive activation are found in other slices as well.



(a) True Activation (b) NMI Method (c) SRA Method

Figure 4.12: Slice 18 of the activation masks for a motion-corrupted dataset. The highlighted voxels exhibit high correlation (both positive and negative) to the stimulus (significance $p = 0.0014$). The activation map in (a) shows the actual, uncorrupted activation, while (b) shows the activation resulting from motion compensation based on the estimates from the normalized mutual information (NMI), and (c) shows the results after motion compensation based on the motion estimates returned by SRA. Similar regions of false-positive activation are found in other slices as well.



(a) (b)

Figure 4.13: False-positive (a) and false-negative (b) activation rates for the standard least-squares (Std), normalized mutual information (NMI), and SRA methods on simulated datasets. Scenario 3 datasets have no false-negative activation because there is no true activation. The error bars show the standard deviation for each measure.

quite different. For reassurance, we include here a comparison between the SRA method and two implementations of the CPI method (CPI-A and CPI-R). We also compare these results to those obtained from two implementations of the unconstrained paired iteration method (UPI-A and UPI-R).

Recall that for both the CPI and UPI methods, we have the choice of starting the iteration with registration or activation detection, and we distinguish between the two by adding a suffix to the acronym: an "R" for registration first, and an "A" for activation detection first.

Our implementation of the paired iteration method is a Perl (Practical Extraction and Report Language) script that makes the appropriate calls to command-line programs of the AFNI (Analysis of Functional NeuroImages) package [20]. AFNI is a suite of highly efficient programs for processing fMRI datasets, and includes almost all the necessary functions required for the paired iteration methods proposed here.

In all of the data processing, only two steps were implemented using custom-designed C++ code. We used an in-house C++ program to combine the motion increment with the current best-guess. This step is necessary because the solution is an iterative process, incrementing the motion estimates at each step. However, this addition does not require a C++ implementation, and can easily be performed using simple Perl functions. Also, we

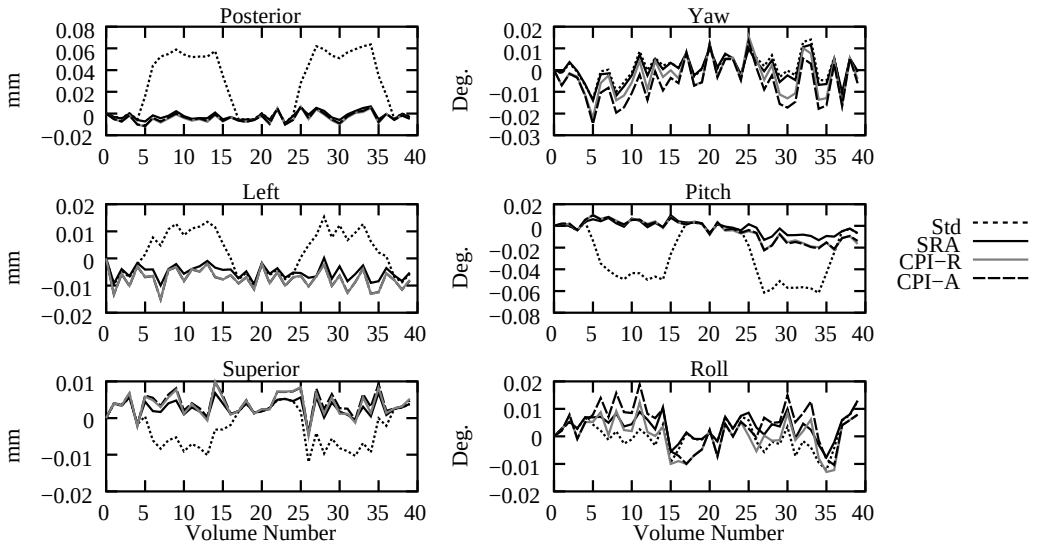


Figure 4.14: Comparison of motion parameter errors for the standard least-squares, SRA, and both CPI registration methods on a simulated dataset that contains activation and random motion. The SRA, CPI-A and CPI-R methods are all very similar.

used a custom C++ program to implement the additional constraints of equations (4.2)-(4.4). The constraint program reads in $\mathbf{A}$ and $\mathbf{B}$, and the initial $(\mathbf{X}, \mathbf{Y})$ solution, and outputs the constrained solution $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$ that minimizes the cost function. The code uses the Nelder-Mead downhill simplex method [71] from *Numerical Recipes in C* [81] to perform the minimization search.

Figure 4.14 plots the errors for the motion estimates returned by the standard, SRA, CPI-R and CPI-A methods. While the standard least-squares registration method shows large stimulus-correlated errors, the three simultaneous methods all return motion estimates that are very close to each other and void of any observable stimulus component. The same plots are shown without the standard method in Fig. 4.15 to allow for a more appropriate axis scaling to compare the simultaneous methods. The largest divergence between any two of the three methods is on the order of 0.005 mm of translation and 0.01° of rotation. These differences are well within the variation observed in the motion plots.

The UPI methods yield results that give an interesting insight into their behaviour, and why these methods fail to return satisfactory solutions. Shown in Fig. 4.16 are the motion plots for the two UPI methods and the standard least-squares registration method for a simulated dataset that contains no activation. As expected, the standard method

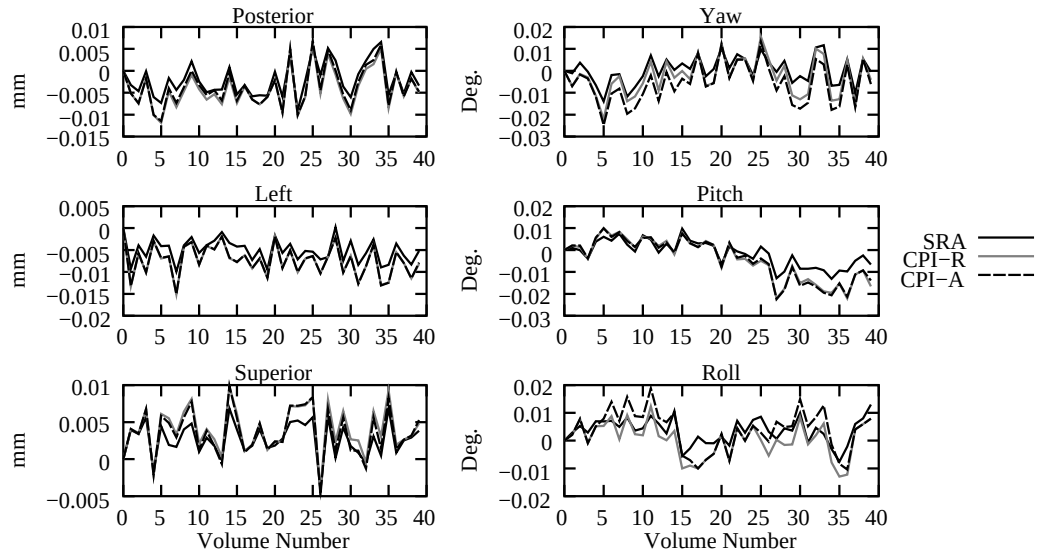


Figure 4.15: Comparison of the SRA method to both CPI registration methods. These graphs show the same data as in Fig. 4.14, but without the standard least-squares method.

returns accurate motion estimates, since there is no activation present to cause an activation bias. The UPI-R method performs equally well. However, the UPI-A method shows very large stimulus-correlated errors. Recall that the unconstrained paired iteration methods do not enforce the minimization of the cost functions outlined in section 4.3, and therefore converge to a solution that may not be appropriate. It makes sense that the eventual solution converged to would depend on the characteristics of the dataset being analysed, and on the processing step done first (registration versus activation detection). On a dataset that has no activation, the UPI-R method yields accurate results because the variation present in the dataset is due to motion and is removed by the initial registration step. The remaining variation is mostly Gaussian noise. As a result, the activation detection finds very little variation that correlates to the stimulus function, and hence makes only a very small adjustment to the dataset when removing the activation. The iterative process simply continues until the registration process converges.

On the other hand, the UPI-A method performs the activation detection first, and does its best to explain the dataset variation as stimulus-correlated activation. As a result, some of the motion variation is accounted for by the activation detection process. After this apparent activation is removed, the resulting voxel time series, minus their average values, are necessarily orthogonal to the stimulus regressors. To achieve this property, linear

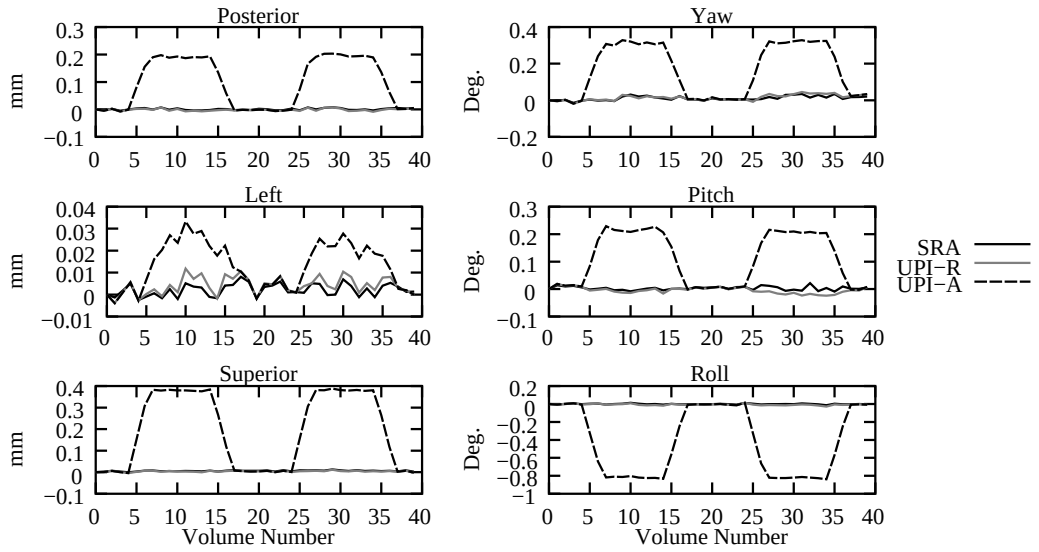


Figure 4.16: Failure of UPI-A method on a simulated dataset that has no activation, but has stimulus-correlated motion.

combinations of the stimulus regressors were added to (or subtracted from) the time series. When the registration algorithm receives the dataset, this variation must then be interpreted as motion. The result is a set of motion parameters with very significant stimulus-correlated components. Note, however, that this is still a least-squares solution to the combined problem. Dataset variation can be passed between the motion and activation components to yield equivalent least-squares solutions, as demonstrated by the derivation in equation (3.14) (on page 55).

A similar argument can be used to justify the results shown in Fig. 4.17, in which motion errors are plotted for the SRA and UPI methods on a simulated dataset containing activation, but no motion. This time, the UPI-A method is accurate, while the UPI-R method exhibits large stimulus-correlated motion errors. Since activation detection is done first in the UPI-A method, the vast majority of the dataset variation is accounted for and removed before the registration step. But in the UPI-R method, the initial registration step tries to account for the BOLD signal as if it were due to motion. As Freire *et al.* pointed out in [30], least-squares registration on a dataset containing a large BOLD signal is susceptible to stimulus-correlated errors. After motion correction using these inaccurate estimates, genuine stimulus-correlated motion is introduced into the dataset by resampling. This motion causes the same type of voxel intensity fluctuations that are responsible for the

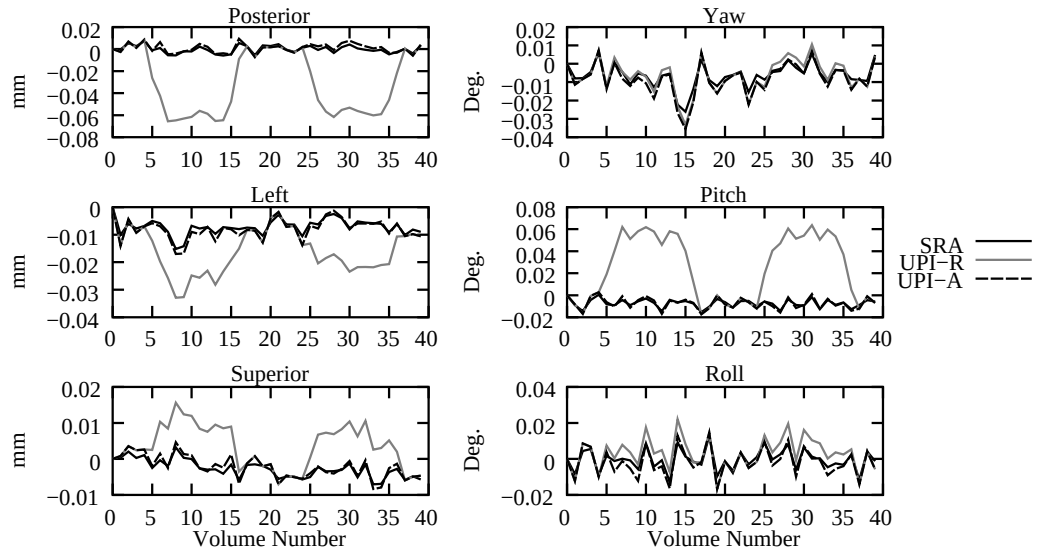


Figure 4.17: Failure of UPI-R method on a simulated dataset that contains activation, but no motion.

false-positive activations reported by Hajnal *et al.* [41]. The false activation, when removed from the dataset, reinforces the stimulus-correlated motion estimates, and the feedback loop continues until it eventually converges to a solution. Again, this least-squares solution to the combined problem is unlikely to be the one desired.

The false-positive and false-negative counts for the seven analysis methods are shown in Tables 4.2 and 4.3, respectively. Notice that the SRA and CPI methods consistently exhibit very similar counts, all resulting in approximately two-thirds fewer false-positive and one-third fewer false negative activations than the standard least-squares algorithm on the datasets that contained activation. Each of the two UPI methods fail to give satisfactory results in at least one of the four dataset scenarios.

4.6 Two Stimulus Conditions

Functional MRI experiments can contain more than one stimulus condition. For example, in an experiment conducted by Bandettini *et al.* [8], a subject was given different finger-tapping paradigms for the right and left hands. The subject was instructed to start and stop tapping with their right hand when cued, and likewise for their left hand. However, the frequency of the on-off switching was 0.08 Hz for the right hand and 0.05 Hz for the

Table 4.2: False-positive counts for the seven algorithms, averaged over 10 trials for each dataset scenario. The true activation mask contains 3454 active voxels.

Dataset Characteristics	Std	NMI	SRA	CPI-A	CPI-R	UPI-A	UPI-R
act., random motion	571.5	684.2	186.3	188.4	187.0	3,877.4	747.5
act., stim-corr motion	617.9	802.2	205.6	203.0	203.9	14,932.1	796.9
no act., stim-corr motion	42.7	125.6	38.4	36.9	36.5	15,732.8	36.8
act., no motion	586.0	785.8	187.8	190.1	188.5	190.2	744.7

Table 4.3: False-negative counts for the seven algorithms, averaged over 10 trials for each dataset scenario. Results for the datasets without activation are not shown because those datasets cannot, by definition, have any false negatives. The true activation mask contains 3454 active voxels.

Dataset Characteristics	Std	NMI	SRA	CPI-A	CPI-R	UPI-A	UPI-R
act., random motion	643.3	616.9	427.9	429.8	435.5	452.0	605.8
act., stim-corr motion	642.3	608.7	429.0	427.0	425.3	379.9	603.6
act., no motion	651.1	627.1	432.3	422.1	431.8	421.0	611.1

left. It was shown that these two frequencies could be resolved in the resulting activation map, allowing the association of voxels to one, both, or neither of the stimuli.

The derivation of the solution to the simultaneous registration and activation detection least-squares problem is general enough to apply to datasets with more than one stimulus regressor. The mathematics is entirely the same. However, the additional constraint must be minimized for each stimulus regressor, each solving for a single column of α. Since there is no coupling between these minimizations, they can be solved for independently.

Simulated datasets containing two stimulus conditions were created in a manner similar to the single-condition datasets outlined in section 4.1. The stimulus function and activation template used in the previous datasets will be referred to as "stimulus 1". A second stimulus function and associated activation template will be referred to as "stimulus 2". The region of activation for stimulus 2 is in the prefrontal cortex, and occupies approximately 8% of the total brain volume. The two stimulus functions are shown in Fig. 4.18 (as well as in the pitch angle plot of Fig. 4.19). It is worth pointing out that the stimulus functions are not simply opposites of each other. The least-squares solution of the activation detection

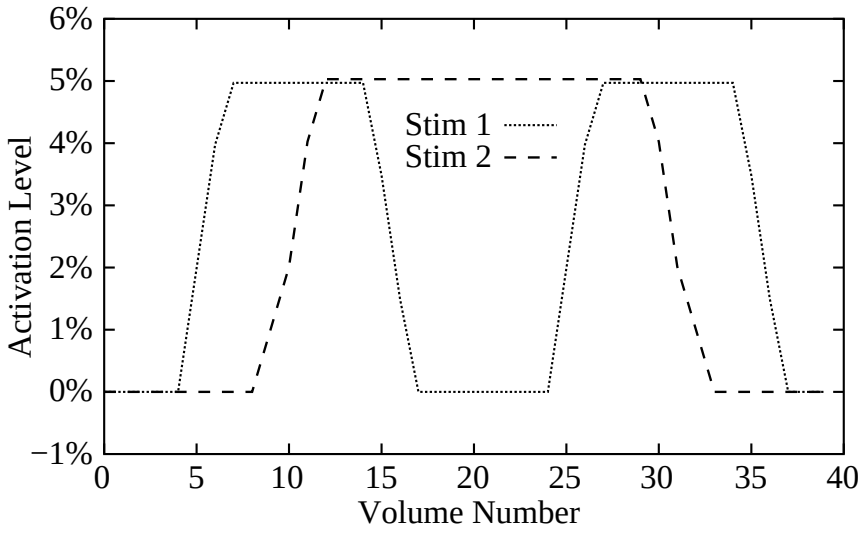


Figure 4.18: Stimulus functions for simulated datasets with two conditions.

problem requires that the set of regressors (rows of $\mathbf{B}$) be linearly independent. If they are not, then the matrix ($\mathbf{BB}'$) will be singular, and the solution process breaks down. This is not a problem with the algorithm, but is the result of a poorly designed fMRI experiment.

Figure 4.19 plots the motion errors of the standard and SRA methods on a simulated dataset that contains both stimulus conditions 1 and 2, as well as motion that is correlated to both stimulus functions. The standard least-squares method exhibits obvious activation bias. The motion estimates returned by the SRA method do not show this bias.

One might ask what would happen if the SRA method was run on a single-condition dataset, but tasked to look for activation from both conditions. To answer this question, the SRA method was given stimulus regressors for both stimuli, but was run on simulated datasets that contained only stimulus 1. In every case, the resulting motion parameters were almost identical to those obtained from the one-condition analysis. Furthermore, the false-positive and false-negative counts were similar to those observed using one stimulus regressor. These results are promising because it demonstrates that the SRA method can be trusted to properly handle datasets in which a particular activation may or may not be present in the data. Furthermore, it opens up the possibility that one regressor can be used to measure activation, while another regressor can be used to model a physiological signal such as respiration or cardiac pulsation.

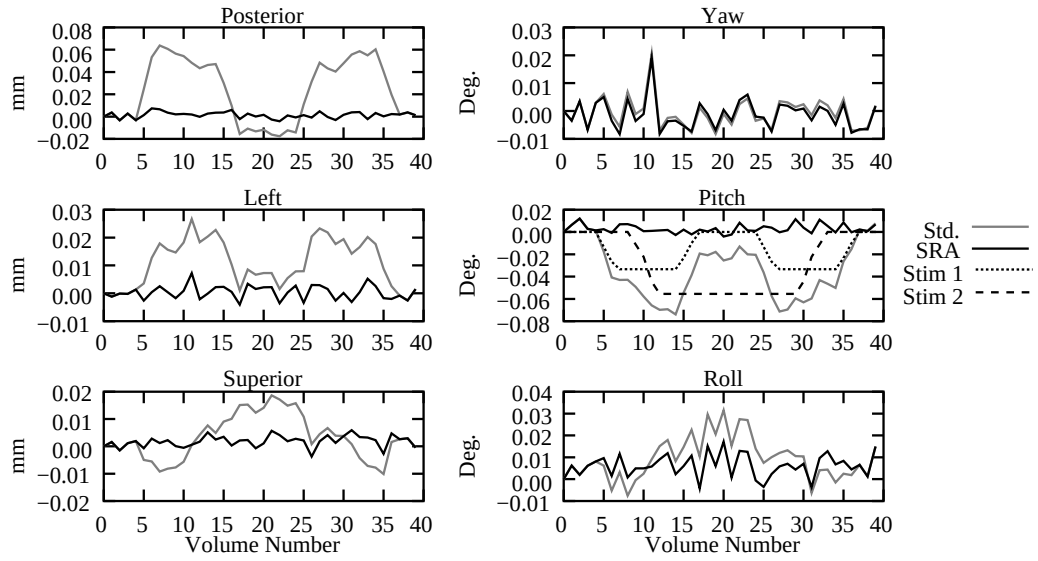


Figure 4.19: Motion estimate errors for the standard and SRA methods on a dataset containing two stimulus conditions. The pitch angle graph also shows the two stimulus functions, each scaled (and negated) to point out that the errors in the parameter estimates of the standard method resemble a linear combination of the two.

4.7 *In Vivo* Data

The SRA method was also tested on four *in vivo* fMRI datasets from different patients and different scanners. Datasets I through III are EPI scans of 180 volumes each[1], acquired at two-second intervals on a 3 T Bruker scanner. Each volume consists of 30 slices (4 mm thick), each having 64×64 voxels (3.75×3.75 mm). The subject was presented with two alternating visual stimuli, each lasting for nine frames. Dataset IV is an EPI scan containing 60 volumes acquired at three-second intervals on a Philips 3 T scanner. Each volume consists of 24 slices of 4-mm thickness. The slices have dimensions 128×128 (1.719×1.719 mm). The stimulus alternated between a visual and motor task, each lasting for 10 frames.

The standard least-squares registration method and the SRA method were both used to analyze the *in vivo* datasets, using the first frame of each dataset as the base frame. For these datasets, the intensity value of 80 was used as an approximate separator for the active and non-active voxels in the activation map, yielding a c-value of 0.05 (see section 4.3)).

[1]In total, 192 volumes were acquired, but the first 12 were discarded because the dipole spins had not yet reached a steady state.

Unlike the simulated datasets, the true motion is unknown for the *in vivo* datasets. As a result, we cannot calculate the error for the motion profiles. Instead, we try to remove any slow subject motion, and focus on the faster transients present in the motion profiles. The motion parameter time-curves were low-pass filtered using a one-stimulus-period windowed averaging (over 18 frames for datasets I, II and III, and 20 frames for dataset IV). These smooth motion trend curves were then subtracted from the motion parameter curves to reveal the detrended curves. The detrended motion parameter curves were then compared to the stimulus function by way of correlation coefficient.

Improved differentiation between motion and activation is evident in results obtained for all *in vivo* datasets. Figure 4.20 plots the standard and SRA motion estimates for dataset II, along with the associated stimulus function. Also shown in the Figure (as well as in Table 4.4) are the correlation coefficients of the detrended motion parameter curves with the stimulus function. In particular, the pitch angle estimates returned by the standard approach are visibly correlated to the stimulus (correlation coefficient of 0.735), while they are much less so in the estimates returned by the SRA method (correlation coefficient of 0.170). The posterior translation parameter exhibits some oscillation for both methods. At first, this variation appears to be stimulus-correlated activation bias. However, the oscillation has at a slightly different frequency than the stimulus function, suggesting that it is mostly due to genuine subject motion.

The magnitude of the overall motion is much larger in dataset IV than in dataset II, as shown in the motion plots in Fig. 4.21. The corresponding stimulus function is shown in the pitch angle plot. Again, the subject motion seems to contain some motion related to the stimulus, although it is not strictly stimulus-correlated motion.

The motion parameters estimated by the SRA method have a lower correlation coefficient with the stimulus than those estimated by the standard method. The correlation coefficients for all four *in vivo* datasets are listed in Table 4.4. An inflated correlation coefficient due to activation bias almost certainly translates directly into an inflated correlation coefficient for a voxel that rests in a nonhomogeneous region of the volume. For example, if the pitch angle contains activation bias, then motion compensation using this erroneous pitch angle will cause voxels (especially those far away from the centre of rotation) to move in synchrony with the stimulus. If the direction of motion happens to follow an intensity level set (i.e. it follows a path of constant intensity), then the voxel's correlation coefficient will not be affected. However, this is often not the case, and many voxels move between positions that

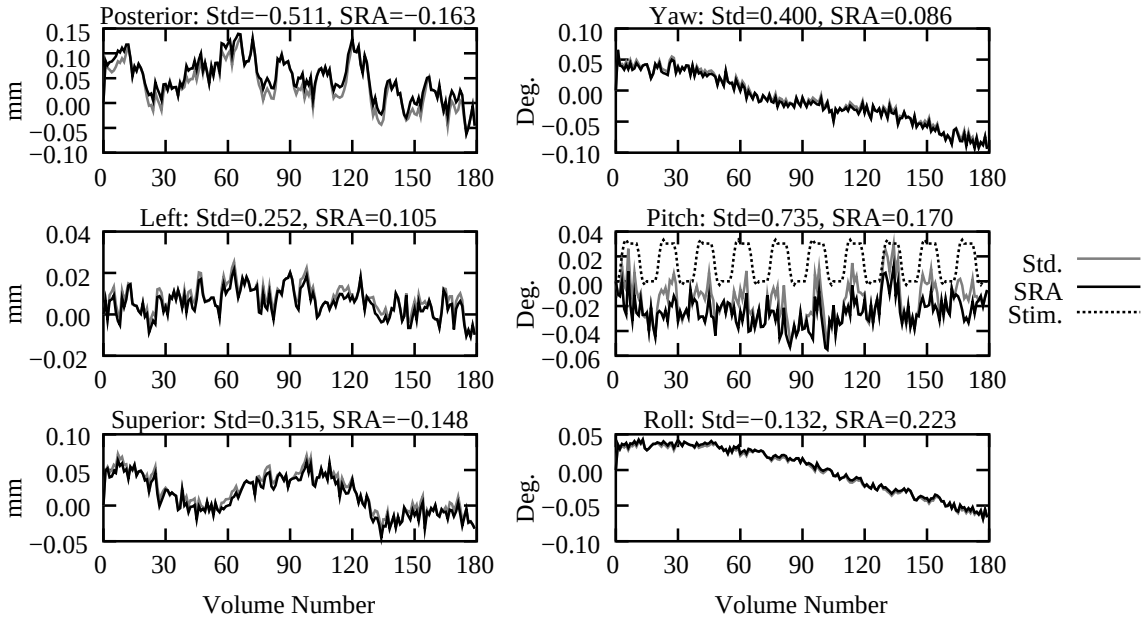


Figure 4.20: Motion estimates for *in vivo* dataset II. The corresponding correlation coefficients are listed above each graph. The stimulus function is also shown in the pitch angle graph.

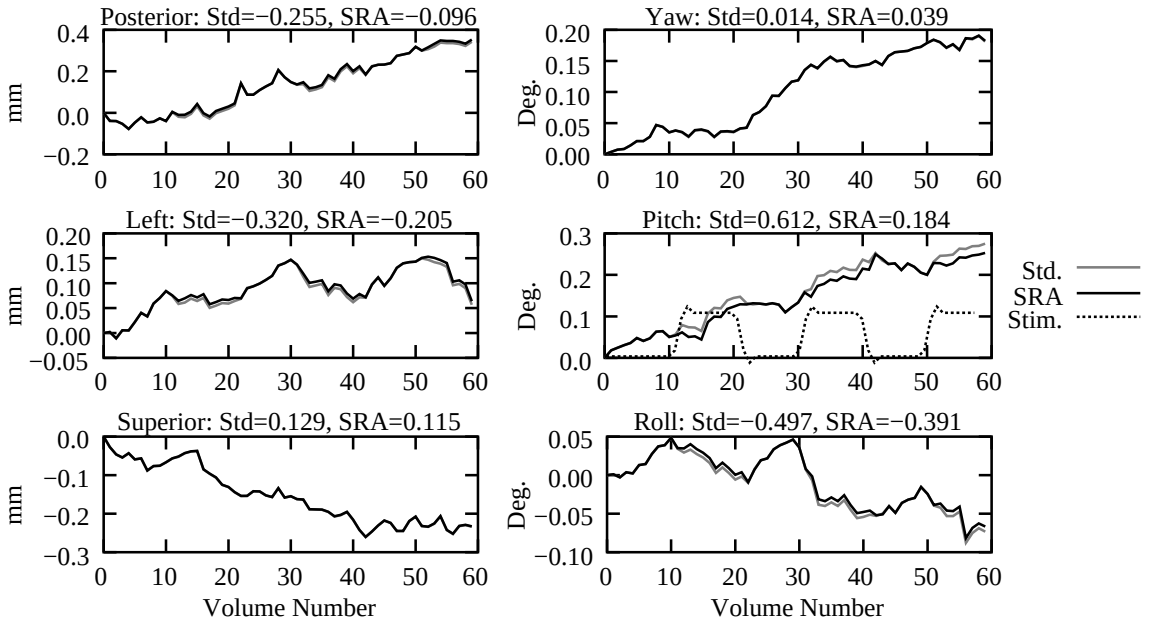


Figure 4.21: Motion estimates for *in vivo* dataset IV. The corresponding correlation coefficients are listed above each graph. The stimulus function is also shown in the left-right translation graph. Note that in some graphs the SRA line obscures the Std. line.

Table 4.4: Correlation coefficients of the detrended motion parameter plots with the corresponding stimulus function.

	Dataset I		Dataset II		Dataset III		Dataset IV	
Parameter	Std	SRA	Std	SRA	Std	SRA	Std	SRA
Posterior	-0.492	-0.055	-0.511	-0.163	-0.467	-0.050	-0.255	-0.096
Left	0.241	0.053	0.252	0.105	0.380	0.077	-0.321	-0.205
Superior	-0.142	-0.314	0.315	-0.148	0.415	0.009	0.129	0.115
Yaw	0.125	-0.031	0.400	0.086	0.305	0.030	0.014	0.039
Pitch	0.724	0.086	0.735	0.170	0.694	-0.004	0.612	0.184
Roll	-0.033	0.125	-0.132	0.223	-0.336	0.067	-0.497	-0.391

differ in intensity, creating the illusion that the voxel's intensity is changing as a result of activation. The fact that the SRA method produces motion estimates that have a lower correlation coefficient than the standard method demonstrates its ability to avoid false-positive activations. It is worth pointing out that the reduction in the motion parameters' correlation to the stimulus function is due to the combined model as well as the constraint on the activation maps. There is no explicit consideration of the motion parameters themselves. The fact that the motion parameters still show a decreased correlation to the stimulus is promising indeed.

4.8 Performance

Not only are the simultaneous methods accurate, but the algorithms are efficient. The following execution-time results were obtained using a desktop workstation with a 1.2 GHz AMD Athlon processor and 512 MB of DDR RAM.

Recall that the simulated datasets each had 40 volumes of dimension $64 \times 64 \times 30$. On these datasets, the standard least-squares registration method typically converged in between 2 and 10 iterations, taking between 1 and 4 minutes. The convergence criterion used was that the motion parameter increment with the largest absolute value had to be smaller than 0.001 (mm or degrees). The SRA method converged in a similar number of iterations, but had the additional task of enforcing the constraint on the resulting activation map, and generally took between 25% and 60% longer to run than the standard method. The Nelder-Mead downhill simplex optimization method used by the SRA algorithm typically

required between 150 and 250 evaluations of the cost function per iteration of the SRA method.

The *in vivo* datasets I, II and III are considerably larger than the simulated datasets. On these, the standard method typically converged in 4 or 5 iterations, taking about 5 minutes in total. In total, the SRA method required a similar number of iterations, and took about twice as long to run. The fact that the SRA method is relatively slower on these datasets than on the simulated datasets is intriguing. The simulated datasets and the *in vivo* datasets have the same number of voxels per volume. Also, each iteration of both the standard and SRA methods is linear in the number of voxels in the dataset. Furthermore, the Nelder-Mead method required between 150 and 250 evaluations of the cost function for each iteration of the SRA method. Both simulated and *in vivo* datasets were analyzed with only one stimulus regressor, so should be subject to similar overheads caused by enforcing the additional constraint. The fact that the SRA method seems to be slightly slower on the *in vivo* datasets is likely due to memory caching issues. The *in vivo* datasets have many more frames than the simulated datasets (compare 180 frames to 40 frames), and take up more memory.

One of the most costly operations required by these registration methods is resampling. To attain the best possible accuracy, all the least-squares methods tested here use Fourier interpolation to apply the rigid-body transformation. This method is slower than linear interpolation because it requires the Fourier transform of various parts of the volume data at several stages (see appendices B.1 and B.2 for a description of the underlying Fourier resampling techniques).

The paired iteration methods were implemented mainly using calls to AFNI [20] programs. In particular, the AFNI program `3drotate` was used to resample the datasets. This program is exceptionally efficient, and results in a substantial speed-up. However, it proved to be quite difficult to attain the same convergence behaviour as the standard and SRA methods. The paired iteration methods were often unable to satisfy the convergence threshold used by the standard and SRA methods (0.001 mm or degrees). Instead, the convergence threshold was set to 0.009 (mm or degrees), still much smaller than the size of the stimulus-correlated errors yielded by the standard least-squares method. It is worth noting that most registration methods operate on only two volumes at a time, and terminate as soon as the motion increments are smaller than the chosen threshold. Different volumes converge in a different number of iterations. However, in the current framework, all the

volumes are registered at the same time, and the iteration does not terminate until *all* the frames return a motion increment smaller than the threshold in the *same* iteration. This is a more restrictive convergence criteria.

On the simulated datasets, both CPI methods typically converged in 2 to 4 iterations, taking between 45 seconds and 2 minutes. The speed-up was due to AFNI's fast resampling, as well as the relaxed convergence criteria. The Nelder-Mead minimization of the additional constraint still required between 150 and 250 evaluations of the cost function. Results of the CPI methods on the *in vivo* datasets showed a similar speed-up, taking about 2 to 3 minutes to converge (with about 3 iterations).

While the paired iteration methods require many files to be written to the disk and read from the disk, the operating system (Linux kernel 2.4.18) was able to cache most of the data in system memory, and avoid the actual physical disk access.

Chapter 5

Conclusions

5.1 Summary

Registration using the standard least-squares method or the normalized mutual information method results in motion estimate errors that are correlated to the BOLD signal in fMRI experiments. As MR scanners of higher field strengths become more commonly used for fMRI, the predominance of these stimulus-correlated mis-registrations can be expected to increase. Using a stimulus-contaminated motion estimate to resample a dataset has the potential to give inactive voxels the appearance of being active, resulting in the false detection of regions of activation. However, our simultaneous registration and activation algorithm accounts for the intensity changes due to the BOLD signal, and minimizes the sum of squares cost function for both the registration and activation least-squares problems in parallel. In simulation studies, our SRA method removes, or greatly reduces, stimulus-correlated motion errors, resulting in a dramatic decrease in the number of false-positive and false-negative activations.

The SRA algorithm is very robust, and works properly on datasets that contain various combinations of stimulus-independent motion, authentic stimulus-correlated motion, and activation. Even patient motion that is in-step with the stimulus does not confound the SRA method's ability to decouple motion effects from activation effects. The result is accurate registration despite any actual confluence between patient motion and the stimulus. This is an impressive feat that other methods are not able to accomplish. Some methods have been proposed to deal with residual motion artifacts that remain after a first pass of registration [10, 35, 40]. In general, their approach is to remove the motion-related

components from each voxel time course. However, when patient motion is authentically stimulus-correlated, these methods have difficulty distinguishing the true BOLD signal from motion artifacts, and run the risk of discounting genuine activation.

Our simultaneous method can also incorporate experimental paradigms with multiple stimulus regressors. In addition, we have shown that including extraneous regressors in the analysis (for which no activation is present in the dataset) does not reduce the method's effectiveness.

Furthermore, our method is generalizable to scaled least-squares registration, in which a seventh parameter is added to the motion detection cost function. This parameter incorporates global intensity scaling [2] and can be used to adjust for global intensity changes.

The SRA method is also relatively easy to implement. Almost all the necessary processing can be offloaded to standard third-party software packages. The only step that we required C++ code to implement was the minimization of the additional constraint. However, even that program consisted mostly of the Nelder-Mead downhill simplex method code from *Numerical Recipes in C* [81]. The remainder of the code is quite simple.

In addition to being robust and reliable, the SRA algorithm is very efficient. Even though the method, by nature, involves the consideration of all voxels from all time steps simultaneously, it is computationally inexpensive and its memory requirements are modest. Analysis of the $64 \times 64 \times 30 \times 180$ voxel *in vivo* datasets was completed in less than four minutes using the CPI implementation. Moreover, there is never a need for the entire dataset to be in memory at once. The paired iteration methods utilize separate registration and activation detection programs, neither of which require the whole dataset to be in memory. Registration needs only two volumes at a time, and activation detection needs only one voxel time series at a time. Any dataset on which these two programs can be run can also be processed using the paired iteration methods.

Considering the superior results of the SRA method, the stringent convergence criteria, and the fact that the SRA code can be further optimized and parallelized, the overall performance of the method is impressive and promising.

5.2 Future Work

The results obtained by our SRA method on *in vivo* datasets are promising. However, for validation, further *in vivo* studies need to be carried out on high-field fMRI datasets for

which the motion is known. One way to acquire such data is to hold the patient still. This sounds simple, but movements as small as 0.05 mm or 0.05° can cause false activations. Even an experienced and cooperative subject cannot consistently stay still to within these standards. More constraining head wraps or clamps might make motion-free datasets possible, but may also interfere with the cognitive task of the fMRI experiment. Instead of requiring that a patient be motionless, an alternative is to measure the patient's motion accurately using a tracking device. These motion estimates, assuming they are accurate enough, can be used as the true motion, allowing for the calculation of registration *errors* for the methods outlined in this thesis.

The simultaneous registration and activation detection model itself, as stated in chapter 3, is not specific to least-squares optimization and might be solved using other cost functions. However, much of the analysis done in this thesis is specific to the least-squares cost function, and it is not clear how different cost functions will alter the results and analyses. The feasibility of using other cost functions should be investigated.

Including a linear time-series regressor is common practice in fMRI activation detection. In the simultaneous methods, each regressor needs to have an additional constraint in order to find a unique solution to the combined problem. It is not clear that the constraints outlined here for the activation maps will be appropriate. Unlike the activation maps, the spatial component associated with a linear drift regressor might not be sparse. An investigation into an appropriate constraint needs to be undertaken.

The SRA method might also be capable of modeling components from physiological signals. Patient respiration causes slow signal variations. If a respiration time course is used as an additional regressor, the effect of respiration can also be accounted for and removed from the registration and activation detection processes. As with the linear drift regressor, an appropriate additional constraint needs to be established in order to make this technique feasible.

The compatibility of our SRA method with non-rigid registration methods should be investigated. The assumption that motion is rigid does not strictly hold. Non-rigid motion at the neck and base of the brain, as well as warping caused by magnetic field inhomogeneities [19], cause distortions in EPI scans.

Many other types of medical imaging datasets produce a time series of volumes that need to be aligned before a time-series analysis can be done. For example, dynamic positron emission tomography (dynamic PET) acquires a time series of volumes while a radioactive

tracer filters through the patient's body and accumulates at the relevant locations. The volumes must be accurately aligned before a voxel-by-voxel time-series analysis can estimate the desired dynamic parameters. This time series of volumes varies widely in both the level and distribution of intensities. It has been shown that naively registering each volume with a chosen reference volume can result in registration errors, and ultimately affect the parameter estimates [106].

Another example of time-series imaging comes from contrast-enhanced MR angiography. These scan sequences acquire a series of MR images (or volumes) while a bolus of contrast agent is injected into the patient. The presence of the contrast agent in the tissue causes a local decrease in the MR signal. By tracking the time course of a voxel's intensity, information can be determined about the tissue's blood in-flow and out-flow characteristics. Such studies are often done to differentiate between areas of the brain with normal and abnormal blood perfusion [65].

Finally, the task of registering images that have differences also appears in non-medical fields such as remote sensing. For example, a time series of satellite images of a coastline can yield information about the long-term erosion and sedimentation (eg. river deltas), or sea-ice flows. Satellite imaging can also be used to monitor deforestation. Accurate registration of these images enables the calculation of quantitative information. However, the regions of mis-match between the images might confound automatic registration methods. This coupling makes the data a good candidate for a simultaneous solution of the motion and dynamic parameters, similar to the way the SRA method handles high-field fMRI datasets.

Appendix A

Physics of MRI

A.1 Bloch Equation and its Solution

Every atom is made up of protons, neutrons, and electrons. Protons have a vector property that physicists call "spin", denoted by the vector $\vec{I}$. Associated with the spin is another vector quantity called the proton's "dipole magnetic moment", denoted $\vec{\mu}$. A proton's spin and its dipole magnetic moment are related by the equation $\vec{\mu} = \frac{\gamma h}{2\pi}\vec{I}$, where h is Planck's constant (6.626×10^{-34}Js), and γ is the gyromagnetic ratio (with units MHz/T). The unit T stands for "Tesla", which is a unit measuring magnetic field strength. One Tesla is approximately equivalent to $20,000$ times the earth's magnetic field. Another unit for measuring magnetic field strength is "gauss". One Tesla is equal to 10^4 gauss.

Magnetic resonance imaging, as suggested by the name, involves a magnetic field. Clinical MRI scanners typically use a magnetic field with strength ranging from 0.2 T to 4 T (research scanners can be up to 9 T). Each proton's dipole magnetic moment is affected by the local magnetic field, and the two are related by the differential equation

$$\frac{d\vec{\mu}}{dt} = \gamma\vec{\mu} \times \vec{B}_0 \quad , \tag{A.1}$$

where $\vec{B}_0$ is the local magnetic field vector.

Most scanners are designed so that the magnetic field is as close to constant as possible throughout the bore of the magnet. Assuming $\vec{B}_0$ is also constant in time, the steady-state solution for $\vec{\mu}$ to equation (A.1) is precession about $\vec{B}_0$ with frequency $\gamma|\vec{B}_0|$. During their precession, dipoles can either be in the same direction as $\vec{B}_0$ (called *parallel* or *spin-up*), or

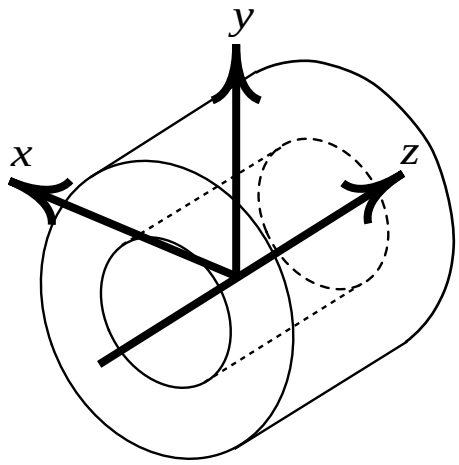


Figure A.1: Scanner coordinate system. The z-axis runs through the bore of the magnet, aligned with the B_0 magnetic field. The y-axis is defined as vertical, and the x-axis is perpendicular to the other two, usually directed from the patient's left to their right side.

pointing in the opposite direction (called *anti-parallel* or *spin-down*). Due to thermal randomization, the dipoles have a slightly greater tendency to be spin-up than spin-down [105]. Although this proportion is small, the sheer number of particles present in a small amount of tissue means that billions more dipoles are spin-up than spin-down.

At this point, it becomes convenient to specify the scanner coordinate system. From this point on, the scanner-fixed reference frame is specified by its x, y and z axes as follows: the z-axis is colinear with $\vec{B}_0$, the y-axis points upward (according to the direction of gravity), and the x-axis is parallel to the ground. Figure A.1 illustrates this coordinate system.

Now consider a small cube of tissue about 1 mm^3 in size. This volume has billions of protons, and hence billions of dipoles. The vector sum of all the dipoles in that small block of tissue is its net magnetization vector, $\vec{M}$. The net magnetization vector can be decomposed into its orthogonal elements, so that $\vec{M} = \vec{M}_x + \vec{M}_y + \vec{M}_z = M_x\vec{i} + M_y\vec{j} + M_z\vec{k}$, where $\vec{i}$, $\vec{j}$ and $\vec{k}$ are the unit axis vectors.

The net magnetization vector is governed by the equation

$$\frac{d\vec{M}}{dt} = \gamma\vec{M} \times \vec{B}_0 - \frac{1}{T_2}\left(M_x\vec{i} + M_y\vec{j}\right) - \frac{1}{T_1}\left(M_z - M_0\right)\vec{k}\ . \tag{A.2}$$

Equation (A.2) is known as the Bloch equation, and is the foundation of MRI [105]. Since $\vec{M}$ is the sum of many dipoles, each precessing about $\vec{B}_0$ but in no coherent phase, the

Tissue	T_1 (msec)	T_2 (msec)
CSF	2400	160
White Matter	780	90
Gray Matter	900	100
Muscle	870	45
Liver	500	40
Fat	260	80

Table A.1: T_1 and T_2 relaxation times for various tissues in a 1.5 T scanner.

steady-state solution of equation (A.2) is $\vec{M}_0$, a vector colinear with $\vec{B}_0$. This is reinforced by the fact that the only equilibrium solution to equation (A.2) is $\vec{M}_0$. In dynamical systems terminology, this solution is known as a globally stable equilibrium solution because all other states eventually converge to it.

The constants T_1 and T_2 represent exponential decay constants for the $\vec{M}_z$ and $\vec{M}_{xy}$ components, respectively, where $\vec{M}_{xy} = \vec{M}_x + \vec{M}_y$. The coefficient T_1 is known as the *longitudinal relaxation time*, or *spin-lattice relaxation time*, since it affects the time evolution of the net magnetization vector component along the longitudinal (z) axis. The coefficient T_2 is called the *transverse relaxation time*, or *spin-spin relaxation time*, because it dictates the rate of decay of the transverse (xy) component of the net magnetization vector. Different tissues have different T_1 and T_2 values, depending on the molecular structure of the tissue's environment.

Thermal jostling of molecules disperses energy. In the presence of heavy molecules such as large protein molecules, this energy dispersal is quite rapid. For this reason, tissues such as fat, and those found in organs like the liver and kidneys, have short T_1 times. Dipoles in light, mobile molecules tend to hang onto this energy longer. Materials such as cerebrospinal fluid (CSF) and water have relatively long T_1 times.

The decay of the transverse component is mostly dependent on local magnetic field inhomogeneities. Dipoles precessing at slightly different frequencies interfere with each other, and can exchange energy. The spins quickly become dephased. If the molecules can move around, like in water, the net magnetic field that a particular molecule experiences is averaged and therefore the magnetic field inhomogeneities are somewhat smoothed out. For this reason, the T_2 time of aqueous materials is longer than that of fixed or elastic tissues. Table A.1 shows the T_1 and T_2 relaxation times for some typical tissues in a 1.5 T scanner.

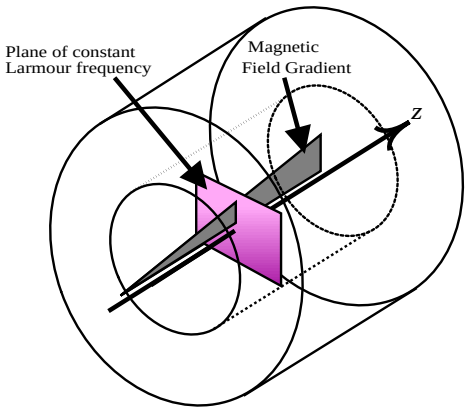


Figure A.2: Slice selection excites a plane of constant Larmour frequency in the presence of a magnetic field gradient along the z-axis.

A.2 Magnetic Field Gradients

As noted in equation (A.1), the frequency at which the dipole magnetic moments precess is proportional to the magnetic field strength. This frequency, called the *Larmour frequency* and denoted ω_0, can be calculated by the formula $\omega_0 = \gamma B_0$, where $B_0 = |\vec{B}_0|$. By passing current through electrical coils in the scanner, gradients in the magnetic field strength can be created. These gradients have a much smaller magnitude than B_0. A typical scanner can create a gradient of a few hundred gauss from one end of the scanner to the other. Compare this to a typical static field strength of 1.5 T or $15,000$ gauss.

A.3 Slice Selection

In many MRI scan sequences, a gradient along the scanner's z-axis is added to the static magnetic field. By doing so, the associated Larmour frequency also has a gradient along the z-axis (since Larmour frequency is proportional to local magnetic field strength). Thus, planes of tissue normal to the z-axis will have a constant Larmour frequency, as shown in Fig. A.2.

Not only do the dipole magnetic moments depend on the local magnetic field, but they also react to radio frequency (RF) electromagnetic radiation. The subject in the scanner is also inside, or next to, a device called a *coil*. The coil is a series of wires designed to

emit RF pulses at specified frequencies. The frequency composition of the RF pulse can be controlled by the scanner. These RF waves are used to excite tissue by disturbing their net magnetization vectors, displacing them from equilibrium. The RF pulse induces a weak spinning magnetic field. It has no effect on dipoles whose Larmour frequency does not match any of the frequencies in the RF pulse. However, those dipoles whose Larmour frequency matches the RF pulse frequency experience a magnetic field that is fixed with respect to their dipole magnetic moments. The RF pulse acts like a torque on a spinning gyroscope, and over time the dipoles begin to tip over and resonate. This causes the net magnetization vector to tip as well, pushing it out of alignment with the static field. This phenomenon is known as *nuclear magnetic resonance*, and has the same properties as other forms of resonance. If a tuning fork is struck close to a guitar that is tuned to its frequency, at least one of the guitar strings will start to vibrate. One can also compare it to pushing a pendulum. If you apply small forces at random, the net effect is that some pushes speed the pendulum up and some slow it down. If you push at the pendulum's natural frequency, the forces add up and the pendulum starts to swing.

When dipoles resonate, the net magnetization vector rotates out of alignment with $\vec{B}_0$, and thus has a growing transverse component, $\vec{M}_{xy}$, and a shrinking longitudinal component, $\vec{M}_z$. This procedure for "exciting" a slice of tissue is known as *slice selection*. Once a slice has been excited, the z-gradient is turned off, and the scanner is ready for the next phase of the imaging process.

A scanner with three gradient coils can create a gradient in any direction. Under the influence of a gradient $\vec{G}$ in the magnetic field strength, equation (A.2) can be applied to tissue at location $\vec{x}$, giving

$$\frac{d\vec{M}}{dt} = \gamma\vec{M} \times \left(\vec{B}_0 + \vec{G}\cdot\vec{x}\right) - \frac{1}{T_2}\left(M_x\vec{i} + M_y\vec{j}\right) - \frac{1}{T_1}\left(M_z - M_0\right)\vec{k}\ . \tag{A.3}$$

Equation (A.3) can be written in matrix form as

$$\frac{d\vec{M}}{dt} = -\begin{bmatrix} \frac{1}{T_2} & -\gamma\left(B_0 + \vec{G}\cdot\vec{x}\right) & 0 \\ \gamma\left(B_0 + \vec{G}\cdot\vec{x}\right) & \frac{1}{T_2} & 0 \\ 0 & 0 & \frac{1}{T_1} \end{bmatrix}\vec{M} + \frac{1}{T_1}\vec{M}_0\ . \tag{A.4}$$

It should be noted that the gradient vector is assumed to have no z-component, hence

$$\vec{G} = G_x\vec{i} + G_y\vec{j}\ . \tag{A.5}$$

The transverse and longitudinal components of $\vec{M}$ are completely decoupled in (A.4). It is more convenient to consider these components separately. First of all, the differential equation corresponding to the z-component is

$$\frac{dM_z}{dt} = \frac{M_0 - M_z}{T_1} \ . \tag{A.6}$$

The general solution to (A.6) is a simple exponential decay toward the equilibrium value of M_0, and is given by

$$M_z(t) = M_0 \left(1 - c_1 e^{-\frac{t}{T_1}}\right) \ , \tag{A.7}$$

where c_1 is an arbitrary constant of integration (used to incorporate an initial condition). The differential equation corresponding to the transverse component of $\vec{M}$ in (A.4) is

$$\frac{d}{dt}\begin{bmatrix} M_x \\ M_y \end{bmatrix} = -\begin{bmatrix} \frac{1}{T_2} & -\gamma\left(B_0 + \vec{G}\cdot\vec{x}\right) \\ \gamma\left(B_0 + \vec{G}\cdot\vec{x}\right) & \frac{1}{T_2} \end{bmatrix}\begin{bmatrix} M_x \\ M_y \end{bmatrix} \ . \tag{A.8}$$

From dynamical systems theory, the general solution to (A.8) is a decaying oscillation with exponential decay constant T_2^{-1} and frequency $\gamma(B_0 + \vec{G}\cdot\vec{x})$. To derive this solution, let us first simplify the notation by letting $V = -T_2^{-1}$ and $W(t) = \gamma(B_0 + \vec{G}\cdot\vec{x})$. Note that $W(t)$ is a function of time because the gradient, $\vec{G}$ can change in time and thus should technically be $\vec{G}(t)$. It is also convenient to think of M_{xy} as a vector in the complex plane. In that frame of reference, $M_{xy}(t) = M_x(t) + iM_y(t)$. Notice again the explicit dependence on t. Then, (A.8) can be written

$$\frac{dM_x(t)}{dt} + i\frac{dM_y(t)}{dt} = \left(V - iW(t)\right)\left(M_x(t) + iM_y(t)\right) \ . \tag{A.9}$$

Expanding the brackets on the right-hand side confirms that the real and imaginary parts coincide with the first and second components of the vector on the right-hand side of (A.8). Another way of representing a complex number is by splitting it into its magnitude and phase. Thus, we will introduce the functions $m(t)$ and $\theta(t)$, both real-valued, such that

$$M_x(t) + iM_y(t) = m(t)e^{i\theta(t)} \ . \tag{A.10}$$

The differential operator on the left-hand side of (A.9) then becomes

$$\frac{dM_x(t)}{dt} + i\frac{dM_y(t)}{dt} = e^{i\theta(t)}\frac{dm(t)}{dt} + im(t)e^{i\theta(t)}\frac{d\theta(t)}{dt} \ . \tag{A.11}$$

After substituting this into (A.9), we can divide the entire equation by $e^{i\theta(t)}$. Expanding the brackets and then collecting the real and imaginary parts leaves us with

$$\frac{dm(t)}{dt} + im(t)\frac{d\theta(t)}{dt} = Vm(t) - iW(t)m(t) \ . \tag{A.12}$$

Since all the functions involved in (A.12) are real-valued (no imaginary parts), we can split it into a system of two equations: one for the real part, and one for the imaginary part.

$$\frac{dm(t)}{dt} = Vm(t) \tag{A.13}$$

$$\frac{d\theta(t)}{dt} = -W(t) \tag{A.14}$$

Notice that a factor of $m(t)$ was divided out of (A.14). This operation is undefined if $m(t) = 0$. However, the phase equation is irrelevant if the magnitude $m(t)$ is zero, so we can assume that $m(t) \neq 0$ in the phase equation. Both of the above equations are easy to solve. Integrating from 0 to t yields

$$m(t) = c_2 e^{Vt} \tag{A.15}$$

$$\theta(t) = -\int_0^t W(s)ds \ , \tag{A.16}$$

where c_2 is another constant of integration representing the initial magnitude, $m(0)$. Thus, putting $m(t)$ and $\theta(t)$ back together, and substituting the appropriate expressions for V and $W(t)$, we get our solution

$$M_{xy}(t) = c_2 e^{\frac{-t}{T_2}} e^{-i\gamma B_0 t} e^{-i\gamma \int_0^t \vec{G}(\tau)\cdot\vec{x}d\tau} \tag{A.17}$$

$$= c_2 e^{-i\gamma B_0 t} e^{\frac{-t}{T_2}} e^{-i\gamma \int_0^t G_x(\tau)x + G_y(\tau)y d\tau} \tag{A.18}$$

$$= c_2 e^{-i\gamma B_0 t} e^{\frac{-t}{T_2}} e^{-i(k_x(t)x + k_y(t)y)} \ , \tag{A.19}$$

where $k_x(t) = \gamma \int_0^t G_x(\tau)d\tau$ and $k_y(t) = \gamma \int_0^t G_y(\tau)d\tau$. The term $e^{\frac{-t}{T_2}}$ represents the exponential decay of the xy (transverse) component of the net magnetization vector. The term $e^{-i\gamma B_0 t}$ signifies the vector's precession due to the static field, while $e^{-i(k_x(t)x + k_y(t)y)}$ alters the precession according to the history of the gradients G_x and G_y.

The above solutions represent the time course of the net magnetization vector of a small piece of tissue at location $\vec{x}$. The transverse component of this magnetization vector can be picked up by the RF coil (the longitudinal component is not measured because it is overwhelmed by the static magnetic field). The signal that is ultimately measured is the

sum of the transverse components of all the net magnetization vectors in the excited slice. Thus, we can represent the measured signal, $S(t)$, by integrating (A.19) over x and y. Note that while γ and B_0 are assumed to be constant throughout the volume of tissue, T_2 is not. Also, since the constant c_2 emerged after an integration in time, it is only assumed to be constant in time. Thus, it can still be a function of position. Any variable that may be a function of position must be inside the integral. The measured MR signal is then

$$S(t) = \int_y \int_x M_{xy}(t) dx dy \tag{A.20}$$

$$= e^{-i\gamma B_0 t} \int_y \int_x c_2 e^{\frac{-t}{T_2}} e^{-i(k_x(t)x + k_y(t)y)} dx dy \ . \tag{A.21}$$

Since c_2 and T_2 are each functions of (x, y), the term term $c_2 e^{\frac{-1}{T_2}}$ is the magnitude of the transverse component of the net magnetization vector at a particular location (x, y) and time t, and will be denoted $M_t(x, y)$. Thus

$$S(k_x, k_y) = \int M_t(x, y) e^{-i(k_x x + k_y y)} dx dy \ , \tag{A.22}$$

where $S(k_x, k_y)$ is $e^{i\gamma B_0 t} S(t)$.

A.4 Fourier Reconstruction

For those familiar with Fourier theory, (A.22) looks like the continuous Fourier transform of $M_t(x, y)$ (details about the Fourier transform are discussed in appendix B). In fact, this similarity is utilized for the reconstruction of $M_t(x, y)$ from the MR signal $S(k_x, k_y)$. The inverse of the Fourier transform is

$$M_t(x, y) = \int S(k_x, k_y) e^{i(k_x x + k_y y)} dk_x dk_y \ . \tag{A.23}$$

Thus, MR data is not acquired in the spatial domain, but rather in the frequency domain. In MRI, this domain is usually referred to as "k-space" because of the use of the variable k to denote the frequency variables. The imaging process consists of acquiring samples $S(k_x, k_y)$ in k-space in order to evaluate $M_t(x, y)$ over the desired field of view. A standard strategy for doing this is by acquiring a single row of k-space during an RF pulse. Filling k-space requires multiple RF excitations. Figure A.3 shows a common strategy for traversing k-space during a scanning sequence.

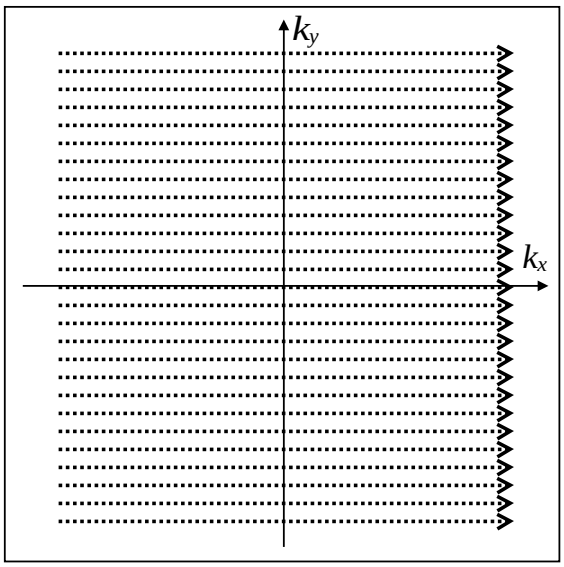


Figure A.3: Typical k-space traversal. One line of k-space is acquired every RF excitation.

Recall that k_x and k_y are defined as

$$k_x = \gamma \int G_x dt \quad \text{and} \quad k_y = \gamma \int G_y dt \ . \tag{A.24}$$

Hence, the traversal through k-space is controlled by changing the x- and y-gradients. Switching on the x-gradient immediately after the RF pulse causes dephasing of the magnetization vectors along the x-axis. Dipoles in the positive x direction will experience a higher magnetic field, and will precess at a higher frequency than those in the negative x direction. As a result, dephasing in the transverse magnetization occurs along the x direction. This dephasing corresponds to changing the phase state, moving along the k_x-axis in k-space.

Immediately after the RF pulse, the signal begins to decay at a high rate. This fast decay is called free induction decay, or FID, and is a combination of T_2 decay and the effect of an imperfect B_0 field [11]. Recall that the signal is the result of many dipoles spinning in phase. If the static magnetic field is perfectly homogeneous, then the only dephasing that will occur is due to T_2 decay. In reality, however, not only is the B_0 field imperfect, but the introduction of an object such as a human being into the scanner causes small changes in the field, especially near high-contrast boundaries (eg. bone/air, or near a metal object). These spatial fluctuations in the magnetic field strength induce small variations in Larmour

frequency. The signal decreases rapidly because these regions of different Larmour frequency quickly go out of phase with each other, and cancel each other out. This type of decay is referred to as T_2^* decay.

For many scan sequences, the effects of the T_2^* decay can be undone using what is called a *spin echo*. A spin echo starts with an RF pulse that rotates the net magnetization vector by 90° and into the transverse plane. At a time $\frac{TE}{2}$ after the 90° RF excitation, another RF pulse is applied, this time rotating the dipoles 180°. This inversion of the dipoles has a refocusing effect. Appropriately, the 180° pulse is often referred to as a *refocusing pulse*. At $\frac{TE}{2}$, the dipoles in one region will be ahead of those in another region. The 180° pulse essentially reverses the direction of all the spins. Then, those spins that were ahead are now behind, and vice versa. After a time $\frac{TE}{2}$ of spinning in the opposite direction, the effects of field inhomogeneity are undone. The spin echo enables accurate, homogeneous T_1- and T_2-weighted imaging. Figure A.4 shows a standard spin-echo scan sequence.

A variation of the spin-echo sequence is the *fast spin-echo* sequence, in which a single 90° RF pulse is followed by repeated 180° RF pulses spaced TE apart, causing the phases to realign repeatedly. This method allows multiple k-space lines to be collected, one line per refocusing pulse. Another version of this type of imaging, called *echo-planar imaging*, is discussed in section A.5.

A.5 Echo-Planar Imaging

The signal change due to the BOLD effect is usually between 1% and 5% of the total signal. Such a low signal change makes it difficult to decipher the meaningful physiological signal from the noise. To overcome this obstacle, many volumes are acquired and jointly analyzed to search for statistically significant changes in the signal. Acquiring many volumes in a short time is a challenge for MRI, but is made possible by a number of specialized scan sequences. One of these types of sequences is called *echo-planar imaging*, or EPI.

Many MRI scanning sequences collect a single line in k-space after each 90° RF excitation. It takes many such excitations to sample k-space sufficiently, so this method can take minutes. Instead of collecting only one line of k-space for each RF excitation, EPI collects many lines at once in a back-and-forth fashion. Some EPI scan sequences collect every second, every third, or every n^{th} line in k-space after an RF pulse, and repeated excitations are

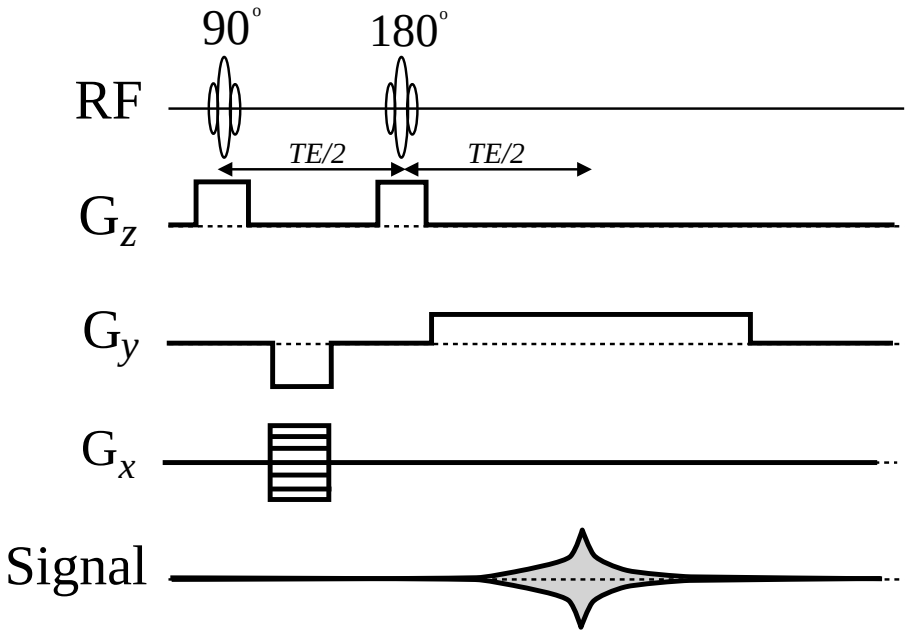


Figure A.4: The spin-echo scanning sequence for a single k-space line is often depicted using a scan sequence diagram like this one. The horizontal axis is time. While the G_z gradient is on, a 90° RF pulse excites a thin slice of tissue. After slice selection, the G_x and G_y gradients are engaged to move the phase state to the point in k-space where the line of samples is to start. The G_x gradient is the same for every line, but the G_y gradient is different for each phase encode. After the correct phase state is achieved, a 180° RF pulse is applied at time $\frac{TE}{2}$. This pulse flips the dipoles. Finally, the G_x gradient is turned on again, and samples are taken as the phase state progresses along the readout line. At time TE, the phases mostly realign, causing an echo.

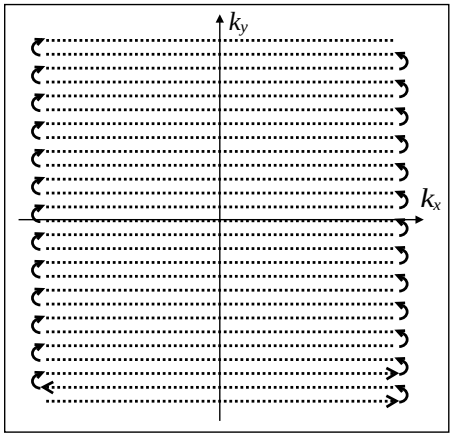


Figure A.5: Single-shot EPI k-space traversal.

needed to fill in all of k-space. This strategy is called *multi-shot echo planar imaging*. However, by sacrificing some resolution or signal-to-noise ratio, an entire slice can be acquired in one RF pulse. This is called *single-shot EPI*, or ss-EPI. This type of high-speed scan cannot be executed by a standard MR scanner. Special gradient coils are needed because the k-space traversal is fast and thus requires very high gradient magnitudes. Furthermore, these gradients have to switch very quickly. Figure A.5 shows the order in which k-space is traversed by a typical EPI scan.

Gradient-echo sequences are similar to spin-echo sequences, except that the echo is induced by reversing the gradients rather than reversing the spins with a 180° RF pulse. In spin-echo MR scans, the echo is created by allowing dephasing to occur, and then reversing all the spins so that those dipoles that were lagging behind are now ahead, and those that were ahead are now behind. After the echo time TE, most of the spins line up.

In gradient-echo MR scan sequences, the concept is slightly different. A gradient is applied, causing spins on one end of the gradient to spin fast, while spins on the other end spin more slowly. After $\frac{TE}{2}$, the gradient is reversed causing the precession frequencies to also reverse; those spins that experienced high Larmour frequencies for the first $\frac{TE}{2}$ now experience low spin frequencies, and vice versa. Finally, after TE, all the spins roughly realign.

The difference between spin-echo imaging and gradient-echo imaging can be illustrated

by the following analogy. Consider two athletes (A and B), and two events (running and crawling). Suppose the two athletes are identical: they run the same speed, and crawl the same speed. Our athletes will each take part in two athletic experiments. Each experiment involves lining up at a starting line on the road, and then either running or crawling along the road.

In the first experiment, called the *spin* experiment, athlete A will run along the road for 5 minutes, turn around, and run back for 5 minutes. Athlete B will do the same, except will be crawling instead of running. Clearly, after 10 minutes both athletes will be back at the starting line. This is analogous to the spin-echo scan sequence. No matter how much faster running is than crawling, the athletes will always end up at the starting line together.

The second experiment, called the *gradient* experiment, involves crawling and running for both athletes. Athlete A will crawl for 5 minutes, then run (in the same direction) for 5 minutes. Athlete B will do the same, except run first, then crawl. After the first 5 minutes, athlete B will be far ahead since running is faster than crawling. However, the roles are reversed for the second half, and the two athletes will have traveled the same distance after 10 minutes. This experiment is analogous to the gradient-echo scan sequence. Notice, however, that one of the major differences between the two experiments is that the spin experiment finished with the athletes at the starting line, while the gradient experiment finished with the athletes a distance away from the starting line that depended on how fast they ran and crawled. This difference is the essence of T_2^* decay.

In terms of imaging contrast, the main difference between EPI images and other anatomical images, such as T_1-weighted or T_2-weighted images, is the magnetic characteristic being measured by the MR signal. Gradient-echo sequences exhibit heavy T_2 and T_2^* contrast.

One of the drawbacks in echo-planar imaging is the suite of problems introduced by fast, imperfect gradient switching. It would be convenient if the gradients could switch instantaneously, creating crisp, well-defined line segments in k-space. However, the reality is that the k-space traversal forms a smooth curve with slightly rounded corners. In spin-echo scan sequences, this issue is of little importance since all the k-space lines are collected in a similar manner and are subject to the same switching error. The consistency makes the error unnoticeable in the final image. Unfortunately, this is not the case in echo-planar imaging because k-space lines are traversed in both directions. Samples are collected at regularly-timed intervals during the scan. Since the time it takes to jump from the end of one k-space line to the beginning of the next is not precisely known, the particular

placement of the samples along the line is also uncertain. The fact that the traversal switches direction means that samples in adjacent lines will be shifted in opposite directions, leading to an imaging phenomenon called *EPI ghosting* [19]. There are methods to deal with and reduce EPI ghosting, but they are outside the scope of this thesis and will not be discussed here [13, 14, 15, 27]. Also, the rapid gradient switching induces *eddy currents*, small fluctuations in the magnetic field [19]. These eddy currents also contribute to timing errors.

Appendix B

Properties of the Fourier Transform

The Fourier transform is a fundamental tool in MRI data processing. As pointed out in appendix A, the Fourier transform and MRI data acquisition are intimately linked. It seems appropriate then that many of the MRI processing operations can be carried out in either the spatial domain (on the image as we view it) or in the frequency domain representation of the image. The frequency domain data can be obtained by taking the Discrete Fourier transform (DFT) of the image, which can be applied to data blocks of any dimension (i.e. 1D, 2D, 3D, etc.).

In most contexts in MRI, the data is defined on a discrete domain. Hence, here the terms "Fourier transform" and "Discrete Fourier transform" will be used interchangeably. Any use of the continuous Fourier transform will be explicit. The DFT of an image or volume $\mathbf{f}$ is often denoted $\mathcal{F}\{\mathbf{f}\}$ or $\hat{\mathbf{f}}$.

The Fourier transform has properties that aid in motion correction image processing. The definition of the DFT for a 2D function $\mathbf{f}_{(m,n)}$ is another 2D function,

$$\begin{aligned} \hat{\mathbf{f}}_{(j,k)} &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \mathbf{f}_{(m,n)} e^{-2\pi i\left(j\frac{m}{M}+k\frac{n}{N}\right)} \\ & \quad j \in [0,\ldots,M-1], \quad k \in [0,\ldots,N-1]\ . \end{aligned} \tag{B.1}$$

B.1 Translation

Translations of $\mathbf{f}$ have a special effect on $\hat{\mathbf{f}}_{(j,k)}$. Consider the DFT of a function $\mathbf{g}_{(m,n)}$ that is simply a translated version of $\mathbf{f}_{(m,n)}$. That is, let $\mathbf{g}_{(m,n)} = \mathbf{f}_{(m-a,n-b)}$, where a and b are the translations along the two axes (in units of voxels). It should be noted that Fourier interpolation assumes a periodic continuation of the function $\mathbf{f}$. That is, for an $M \times N$ image $\mathbf{f}$, we define $\mathbf{f}_{(-m,-n)}$ to be equal to $\mathbf{f}_{(M-m,N-n)}$. The DFT of $\mathbf{f}_{(m-a,n-b)}$ is

$$\begin{aligned}
\hat{g}_{(j,k)} &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_{(m-a,n-b)} e^{-2\pi i\left(j\frac{m}{M}+k\frac{n}{N}\right)} && \text{(B.2)}\\
&= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_{(m,n)} e^{-2\pi i\left(j\frac{m+a}{M}+k\frac{n+b}{N}\right)} && \text{(B.3)}\\
&= e^{-2\pi i\left(j\frac{a}{M}+k\frac{b}{N}\right)} \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_{(m,n)} e^{-2\pi i\left(j\frac{m}{M}+k\frac{n}{N}\right)} && \text{(B.4)}\\
&= e^{-2\pi i\left(j\frac{a}{M}+k\frac{b}{N}\right)} \hat{f}_{(j,k)} \ . && \text{(B.5)}
\end{aligned}$$

This phenomenon, in which translation in the spatial domain is equivalent to multiplication of an exponential function in the frequency domain, is known as the Fourier Shift theorem. Notice that the remaining exponent in equation B.2 is a linear function of j and k. Also notice that the exponential term has a magnitude of 1, since it has no real part. Thus, multiplication by this factor simply applies a phase shift to $\hat{\mathbf{f}}_{(j,k)}$ that linearly depends on the coordinates j and k. The operation is often referred to as applying a *phase ramp*, or *phase modulation*.

The inverse of the DFT is very similar to the DFT itself:

$$\begin{aligned}
f_{(m,n)} &= \frac{1}{MN}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1} \hat{f}_{(j,k)} e^{2\pi i\left(m\frac{j}{M}+n\frac{k}{N}\right)} && \text{(B.6)}\\
&\qquad m \in [0,\ldots,M-1], \quad n \in [0,\ldots,N-1] \ .
\end{aligned}$$

It is simple to show that the inverse DFT (iDFT) also has the Fourier Shift property. Thus, a translation in the frequency domain is equivalent to a phase modulation in the spatial domain. Likewise, a translation in the spatial domain is the same as a phase modulation in the frequency domain. These two reciprocal operations are fundamental in MRI processing.

B.2 Rotation

Before we move on to study the effect of rotations, we must clearly define our coordinate systems. The coordinate system in which (m, n) resides is called the *storage coordinate system*. Its units are set such that a value of 1 represents the distance between samples. Contrast this to the *physical coordinate system*, the space corresponding to the real-world in which the subject exists. While it is convenient to have the physical coordinate system axes aligned with those of the storage coordinate system, the sample spacing in the physical system does not need to be isotropic. For example, an MR image with in-plane dimensions of 128×128 might correspond to a field of view (FOV) of 12.8×25.6 cm. Clearly, the sample spacing is non-isotropic in this case.

When we speak of rotations in MRI processing, we are almost always referring to rotations in the physical coordinate system. Special allowances must be made when applying such a rotation to MRI data that was collected on a non-isotropic sampling grid. The appropriate rotation can be achieved by simply scaling the axes before applying the rotation operation. The effect essentially has storage coordinates trace out ellipses, rather than circles.

Another quirky property of the DFT is that changing the sample spacing in the spatial domain changes the FOV in the frequency domain, and vice versa. This behaviour comes out of the relationship

$$\Omega = \frac{2\pi}{S} \ , \tag{B.7}$$

where S is the (physical system) sample spacing in the spatial domain, and Ω is the FOV in the frequency domain. This relationship is applied to each axis independently. Notice that if the spatial domain is sampled isotropically (equal sample spacing along all axes), then the sampled region of the frequency domain is a square (in the 2D case).

Rotations are invariant under the Fourier transformation. However, to derive this characteristic, we switch to using continuous-domain functions, and the continuous Fourier transform. Let $f(\vec{x})$ be a function on the continuous variable $\vec{x}$ in $\mathbb{R}^2$ or $\mathbb{R}^3$. If we denote a rotation of θ radians by the operator R_θ, then let $f(R_\theta \vec{x})$ be a rotated copy of $f(\vec{x})$. The continuous Fourier transform of $f(R_\theta \vec{x})$ is

$$\mathcal{F}\{f(R_\theta \vec{x})\}(\vec{s}) = \int_{-\infty}^{\infty} f(R_\theta \vec{x}) e^{-i\vec{s}\cdot\vec{x}} d\vec{s} \tag{B.8}$$

$$= \int_{-\infty}^{\infty} f(\vec{x}) e^{-i\vec{s}\cdot\left(R_\theta^{-1}\vec{x}\right)} d\vec{s} \tag{B.9}$$

$$= \int_{-\infty}^{\infty} f(\vec{x}) e^{-i(R_\theta \vec{s})\cdot\vec{x}} d\vec{s} \tag{B.10}$$

$$= \mathcal{F}\{f(\vec{x})\}(R_\theta \vec{s}) \ , \tag{B.11}$$

where the integral sign actually represents integration over a multi-dimensional domain. This derivation shows that rotating the image in the spatial domain by θ degrees is the same as rotating its Fourier transform in the frequency domain by θ degrees. This same concept carries over in an approximate sense to discrete functions. That is, if one rotates the DFT of an image (using resampling), then the iDFT represents an approximation to the rotated image.

Because of the Fourier Shift theorem, and the behaviour of rotations, six degree-of-freedom motion detection can also be carried out in the frequency domain. The feasibility of doing such MRI processing in the frequency domain was first demonstrated by Wood *et al.* [99].

B.3 Convolution

One of the most powerful properties of the Fourier transform is its treatment of convolution. Consider the two discrete functions $\mathbf{f}_j$ and $\mathbf{g}_j$, $j = 0, \ldots, N-1$. Some operations call for the convolution of two functions, such as Gaussian smoothing and interpolation. Their convolution, denoted by $\mathbf{f} \star \mathbf{g}$, is defined as

$$(\mathbf{f} \star \mathbf{g})_j = \sum_{k=0}^{N-1} \mathbf{f}_k \, \mathbf{g}_{j-k} \ . \tag{B.12}$$

Then, the DFT of $(f \star g)$ is

$$\mathcal{F}\{\mathbf{f} \star \mathbf{g}\}_n = \sum_{j=0}^{N-1} \left(\sum_{k=0}^{N-1} \mathbf{f}_k \, \mathbf{g}_{j-k} \right) e^{-2\pi i \frac{jn}{N}} \tag{B.13}$$

$$= \sum_{j=0}^{N-1} \mathbf{f}_k \, e^{-2\pi i \frac{kn}{N}} \sum_{k=0}^{N-1} \mathbf{g}_{j-k} \, e^{-2\pi i \frac{n}{N}(j-k)} \tag{B.14}$$

$$= \hat{\mathbf{f}}_n \, \hat{\mathbf{g}}_n \ . \tag{B.15}$$

It should be noted that the above derivation assumes that $\mathbf{f}$ and $\mathbf{g}$ are periodic. Equations (B.13)-(B.15) show that convolution in one domain corresponds to simple multiplication in the other domain. This means that one can perform a costly convolution filtering in

the spatial domain by multiplying the DFT of the two functions, and performing the iDFT on the result. An application of performing convolution using these methods is discussed in section 2.3.3 (page 35) under the heading "Correlation".

B.4 Fourier Interpolation

It is often essential to estimate the intensity of an image at points that do not coincide with actual samples. The process of estimating the value of a function between samples is called *interpolation* or *resampling*. Standard interpolation techniques are based on forming a continuous version of the dataset by convolving it with a continuous interpolation kernel.

Since convolution is such an expensive operation, interpolation is usually carried out using a kernel that is non-zero on only a small, compact domain. For example, trilinear, windowed sinc, and polynomial interpolation methods all involve only the samples in the neighbourhood of the interpolation point. While a smaller kernel makes the convolution operation faster, it causes aliasing artifacts. To see why this occurs, consider carrying out the convolution in the frequency domain. The operation is simply the multiplication of the Fourier coefficients of the interpolation kernel, and the DFT coefficients of the image. The problem is that the DFT of the image is periodic, and includes not only one copy of the DFT coefficients, but an infinite number of harmonic copies as well. This is not a problem when the interpolation kernel is band-limited so that its Fourtier transform is only non-zero over one period of the image DFT coefficients. But, since the interpolation kernel is non-zero on a bounded domain, its Fourier transform must be non-zero on an infinite domain. That is, the Fourier transform of a compact-domain function has an infinite domain. When performing the multiplication in the frequency domain, the extended domain of the interpolation kernel's Fourier coefficients cause some of the image's harmonics to contribute to the filtered image. This results in aliasing artifacts, a blurring or ringing in the filtered image, as well as a loss of information [40, 42, 70, 78].

Aliasing can be removed by assuming that the interpolation kernel (and therefore the underlying function being sampled) is band-limited[1]. If the interpolation kernel has no frequencies that overlap the image's harmonics, then the interpolation operation does not result in the loss of information, and is invertible.

[1]It should be noted that the actual object being imaged is likely not band-limited. For example, the skin/air interface represents a jump discontinuity in material properties.

Because of these aliasing issues, it is generally accepted that one of the best interpolation methods is Fourier interpolation [40]. Fourier interpolation samples a continuous function based on a discrete function as follows. The DFT of the discrete function is calculated yielding a discrete, frequency-domain image. However, instead of treating the DFT coefficients as one period of a discrete, periodic sequence, we can interpret them as the *only* non-zero portion of a non-periodic sequence. Then, the spatial domain and frequency domain no longer make up a DFT pair. Taking the inverse (continuous) Fourier transform of the non-periodic, discrete frequency data yields a band-limited, periodic function on a continuous domain. The operation of Fourier interpolation is accomplished by shifting this continuous function so that the function is offset with respect to the sample points. Then we can make our way back by reversing the steps. First, we take the (continuous) Fourier transform of the (now shifted) continuous function, yielding the non-zero discrete samples in the frequency domain. A periodic continuation of this data turns it back into one half of a DFT pair. Finally, applying the inverse DFT returns the discrete samples of the (shifted) continuous function.

To make this idea a little more concrete, consider a function, $f(x, y)$, defined on a bounded, continuous domain. The function is sampled during the imaging process. This is essentially what really happens during imaging; a person's anatomy is defined in continuous space, but is sampled to produce a discrete image. Let the discrete, sampled image be denoted $\mathbf{f}_{(m,n)}$. If we assume that the function is periodic, then we can express the continuous function as a bi-infinite series of trigonometric basis functions,

$$f(x, y) = \frac{1}{MN} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \hat{\mathbf{f}}_{(j,k)} e^{2\pi i \left(x \frac{j}{M} + y \frac{k}{N} \right)} , \tag{B.16}$$

where $\hat{\mathbf{f}}_{(j,k)}$ is the infinite series of Fourier coefficients. Performing the inverse continuous Fourier transform returns the original function [75]. However, truncating the series yields a smooth, periodic function in the spatial domain. This truncation operation forces the image to be band-limited. If enough of the Fourier coefficients remain (enough to satisfy the Nyquist criterion), aliasing is avoided because repetitions of the periodic function do not overlap. Fourier interpolation allows one to sample this smooth approximation at any point, not just at the sample points, by shifting the image, or equivalently, the interpolation kernel.

The required Fourier coefficients, $\hat{\mathbf{f}}_{(j,k)}$, are conveniently the same as those given by the

discrete Fourier transform of $\mathbf{f}_{(m,n)}$. Suppose we have the discrete image $\mathbf{f}_{(m,n)}$, and want to use Fourier interpolation to sample the image with an offset of one-half pixel in the x-direction. That is, we want to interpolate our image to get values for $\mathbf{f}_{(m+\frac{1}{2},n)}$. We can do this by first taking the DFT of $\mathbf{f}$ to get the coefficients $\hat{\mathbf{f}}_{(i,j)}$. We then multiply by the phase ramp

$$e^{-2\pi i \frac{j}{2M}} \tag{B.17}$$

to induce the one-half pixel translation in the x-direction (as in section B.1). Simply taking the inverse DFT of

$$e^{-2\pi i \frac{j}{2M}} \hat{\mathbf{f}}_{(j,k)} \tag{B.18}$$

yields the interpolated image. The result is a version of our original image that is shifted by one-half of a pixel.

Fourier interpolation can be used to facilitate accurate rotations by decomposing the rotation operation into a sequence of shears. Shears can be implemented by applying translations to rows of data points, and hence can be done using Fourier interpolation. Rotations using shear transformations for purposes of 2D registration is documented by Eddy [23]. This method was expanded to 3D registration of fMRI volumes by Cox and Jesmanowicz [21]. Although the shear rotations introduce aliasing, the robust interpolation often outweighs their negative effects [29]. Cox *et al.* also point out that satisfactory results can be attained with less computation using 3^{rd}-, 5^{th}- and 7^{th}-degree Lagrange polynomial interpolation. For an explanation of Lagrange polynomials, see Burden and Faires [16] or Anton and Rorres [3].

Bibliography

[1] S. Alliney and S. Ruzisky. An algorithm for the minimization of mixed ℓ_1 and ℓ_2 norms with applications to Bayesian estimation. *IEEE Trans Signal Proc*, 42(3):618–627, 1994.

[2] N. M. Alpert, D. Berdichevsky, Z. Levin, E. D. Morris, and A. J. Fischman. Improved methods for image registration. *NeuroImage*, 3:10–18, 1996.

[3] Howard Anton and Chris Rorres. *Elementary Linear Algebra with Applications*. John Wiley & Sons, 1987.

[4] David Atkinson and Derek L. G. Hill. Automatic correction of motion artifacts in magnetic resonance images using an entropy focus criterion. *IEEE Trans Med Imaging*, 16(6):903–910, 1997.

[5] David Atkinson and Derek L. G. Hill. Automatic motion correction using prior knowledge. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 747, Glasgow, April 2001.

[6] David Atkinson, Derek L. G. Hill, Peter N. R. Stoyle, Paul E. Summers, and Stuart Clare. Automatic compensation of motion artifacts in MRI. *Magn Reson Med*, 41:163–170, 1999.

[7] David Atkinson, David A. Porter, Derek L. G. Hill, Fernando Calamante, and Alan Connelly. Sampling and reconstruction effects due to motion in diffusion-weighted interleaved echo planar imaging. *Magn Reson Med*, 44:101–109, 2000.

[8] Peter A. Bandettini, Andrzej Jesmanowicz, Eric C. Wong, and James S. Hyde. Processing strategies for time-course data sets in functional MRI fo the human brain. *Magn Reson Med*, 30:161–173, 1993.

[9] Peter A. Bandettini, Eric C. Wong, R. Scott Hinks, Ronald S. Tikofsky, and James S. Hyde. Time course EPI of human brain function during task activation. *Magn Reson Med*, 25:390–397, 1992.

[10] Peter R. Bannister, Christian F. Beckmann, Mark Jenkinson, S.M. Smith, and J. M Brady. Motion artefact decorrelation in fMRI analysis using ICA. In *Proceedings of the 10th Annual Meeting of ISMRM*, Honolulu, May 2002.

[11] Jacob Beutel, Harold L. Kundel, and Richard L. Van Metter, editors. *Handbook of Medical Imaging*, volume 1: Physics and Psychophysics. SPIE Press, 2000.

[12] Ake Bjorck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[13] Michael H. Buonocore and Lisheng Gao. Ghost artifact reductionfor EPI using image phase correction. *Magn Reson Med*, 38:89–100, 1997.

[14] Michael H. Buonocore and David C. Zhu. Image-based ghost correction for general interleaved EPI. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 292, Glasgow, April 2001.

[15] Michael H. Buonocore and David C. Zhu. Image-based ghost correction for interleaved EPI. *Magn Reson Med*, 45:96–108, 2001.

[16] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. PWS-Kent, fourth edition, 1989.

[17] A. Caunce and C. J. Taylor. Using local geometry to build 3D sulcal models. In *Proceedings of Info Proc Med Imaging*, pages 196–209, Vancouver, September 1999.

[18] Kai-Hsiang Chuang and Jyh-Horng Chen. IMPACT: Image-based physiological artifacts estimation and correction technique for functional MRI. *Magn Reson Med*, 46:344–353, 2001.

[19] M. S. Cohen. Echo-planar imaging and functional MRI. In C. W. T. Moonen and P. A. Bandettini, editors, *Functional MRI*, pages 137–148. Springer-Verlag, 1999.

[20] Robert W. Cox. AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, 29:162–173, 1996.

[21] Robert W. Cox and Andrzej Jesmanowicz. Real-time 3D image registration for functional MRI. *Magn Reson Med*, 42:1014–1018, 1999.

[22] Robert W. Cox, Andrzej Jesmanowicz, and James S. Hyde. Real-time functional magnetic resonance imaging. *Magn Reson Med*, 33:230–236, 1995.

[23] William F. Eddy, Mark Fitzgerald, and Douglas C. Noll. Improved image registration using Fourier interpolation. *Magn Reson Med*, 36(6):923–931, 1996.

[24] R. L. Ehman and J. P. Felmlee. Adaptive technique for high-definition MR imaging of moving structures. *Radiology*, 173(1):255–263, 1989.

[25] J. Michael Fitzpatrick, Jay B. West, and Calvin R. Maurer, Jr. Predicting error in rigid-body point-based registration. *IEEE Trans Med Imaging*, 17(5):694–702, 1998.

[26] K. Forbes, J. G. Pipe, R. Bird, and J. E. Heiserman. PROPELLER MRI: clinical testing of a novel technue for quantification and compensation of head motion. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 289, Glasgow, April 2001.

[27] David L. Foxall, Paul R. Harvey, and Jian Huang. Rapid iterative reconstruction for echo planar imaging. *Magn Reson Med*, 42:541–547, 1999.

[28] Joseph A. Frank, John L. Ostuni, Yihong Yang, Yoseph Shiferaw, Anand Patel, Jiangning Qin, Venkata S. Mattay, Bobbi K. Lewis, Ronald L. Levin, and Jeff H. Duyn. Technical solution for an interactive functional MR imaging examination: Application to a physiologic interview and the study of cerebral physiology. *Radiology*, 210:260–268, 1999.

[29] Donald Fraser and Robert A. Schowengerdt. Avoidance of additional aliasing in multipass image rotations. *IEEE Trans Med Imaging*, 3(6):721–735, 1994.

[30] Luis Freire and Jean-François Mangin. Motion correction algorithms may create spurious brain activations in the absence of subject motion. *NeuroImage*, 14:709–722, 2001.

[31] Luis Freire and Jean-François Mangin. Motion correction algorithms of the brain mapping community create spurious functional activations. In *Proceedings of Info Proc Med Imaging*, pages 246–258, Davis, June 2001.

[32] Luis Freire and Jean-François Mangin. Two-stage alignment of fMRI time series using the experiment profile to discard activation-related bias. In Takeyoshi Dohi and Ron Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention (MICCAI'02)*, volume 2489 of LNCS, pages 663–670, Tokyo, September 2002.

[33] Luis Freire, A. Roche, and Jean-François Mangin. What is the best similarity measure for motion correction in fMRI time series? *IEEE Trans Med Imaging*, 21(5):470–484, 2002.

[34] Karl J. Friston. Statistical parametric mapping and other analyses of functional imaging data. In Arthur W. Toga and John C. Mazziotta, editors, *Brain Mapping: The Methods*, pages 363–386. Academic Press, 1996.

[35] Karl J. Friston, Steven Williams, Robert Howard, Richard S. J. Frackowiak, and Robert Turner. Movement-related effects in fMRI time-series. *Magn Reson Med*, 35:346–355, 1996.

[36] Zhuo Wu Fu, Yi Wang, Roger C. Grimm, Phillip J. Rossman, Joel P. Felmlee, Stephen J. Riederer, and Richard L. Ehman. Orbital navigator echoes for motion measurements in magnetic resonance imaging. *Magn Reson Med*, 34:746–753, 1995.

[37] Daniel Gembris, John G. Taylor, Stefan Schor, Wolfgang Frings, Dieter Suter, and Stefan Posse. Functional magnetic resonance imaging in real time (FIRE): Sliding-window correlation analysis and reference-vector optimization. *Magn Reson Med*, 43:259–268, 2000.

[38] Gary H. Glover, Tie-Qiang Li, and David Ress. Image-based method for retrospective correction of physiological motion effects in fMRI: RETROICOR. *Magn Reson Med*, 44:162–167, 2000.

[39] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Third Edition, Johns Hopkins University Press, Baltimore, MD, 1996.

[40] S. Grootoonk, C. Hutton, J. Ashburner, A. M. Howseman, O. Josephs, G. Rees, K. J. Friston, and R. Turner. Characterization and correction of interpolation effects in the realignment of fMRI time series. *NeuroImage*, 11:49–57, 2000.

[41] Joseph V. Hajnal, Ralph Myers, Angela Oatridge, Jane E. Schwieso, Ian R. Young, and G. M. Bydder. Artifacts due to stimulus correlated motion in functional imaging of the brain. *Magn Reson Med*, 31:283–291, 1994.

[42] Joseph V. Hajnal, Nadeem Saeed, Elaine J. Soar, Angela Oatridge, Ian R. Yound, and Graeme M. Bydder. A registration and interpolation procedure for subvoxel matching of serially acquired MR images. *J Comput Assist Tomogr*, 19(2):289–296, 1995.

[43] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, 1998.

[44] Derek L. G. Hill and David J. Hawkes. Across-modality registration using intensity-based cost functions. In Isaac Bankman, editor, *Handbook of Medical Imaging: Processing and Analysis*, chapter 34, pages 537–553. Academic Press, 2000.

[45] Derek L. G. Hill, C. Studholme, and David J. Hawkes. Voxel similarity measures for automated image registration. *Proceedings of SPIE Med Imaging*, 2359:205–216, 1994.

[46] Mark Holden, Derek L. G. Hill, Erika R. E. Denton, Jo M. Jarosz, Tim C. S. Cox, Torsten Rohlfing, Joanne Goodey, and David J. Hawkes. Voxel similarity measures for 3-D serial MR brain image registration. *IEEE Trans Med Imaging*, 19(2):94–102, 2000.

[47] Xiaoping Hu and Seong-Gi Kim. Reduction of signal fluctuation in functional MRI using navigator echoes. *Magn Reson Med*, 31:495–503, 1994.

[48] Xiaoping Hu, Tuong Huu Le, Todd Parrish, and Peter Erhard. Retrospective estimation and correction of physiological fluctuation in functional MRI. *Magn Reson Med*, 34:201–212, 1995.

[49] M. Jenkinson and S. Smith. A global optimization method for robust affine registration of brain images. *Med Image Anal*, 5(2):143–156, 2001.

[50] A. K. Katsaggelos, J. Biemond, R. W. Schafer, and R. M. Merserau. A regularized iterative image restoration algorithm. *IEEE Trans Signal Proc*, 39(4):914–929, 1991.

[51] Hope W. Korin, Joel P. Felmlee, Stephen J. Riederer, and Richard L. Ehman. Spatial-frequency-tuned markers and adaptive correction for rotational motion. *Magn Reson Med*, 33:663–669, 1995.

[52] Vadim Kuperman. *Magnetic Resonance Imaging: Physical Principles and Applications*. Academic Press, 2000.

[53] N. Lange. Statistical procedures for functional MRI. In C. W. T. Moonen and P. A. Bandettini, editors, *Functional MRI*, pages 301–335. Springer-Verlag, 1999.

[54] Tuong Huu Le and Xiaoping Hu. Retrospective estimation and correction of physiological artifacts in fMRI by direct extraction of physiological activity from MR data. *Magn Reson Med*, 35:290–298, 1996.

[55] C. C. Lee, R. C. Grimm, A. Manduca, P. Felmlee, R. L. Ehman, S. J. Riederer, and C. R. Jack, Jr. A prospective approach to correct for inter-image head rotation in fMRI. *Magn Reson Med*, 39(2):232–243, 1998.

[56] Christine C. Lee, Clifford R. Jack, Jr., R. C. Grimm, Philip J. Rossman, J. P. Felmlee, R. L. Ehman, and Stephen J. Riederer. Real-time adaptive motion correction in fMRI. *Magn Reson Med*, 36:436–444, 1996.

[57] Christine C. Lee, Clifford R. Jack, Jr., Philip J. Rossman, and Stephen J. Riederer. Real-time reconstruction and high-speed processing in functional MR imaging. *Am J Neuroradiol*, 19:1297–1300, 1998.

[58] Heva Lester, Simon R. Arridge, Kalvis M. Jansons, Louis Lemieux, Joseph V. Hajnal, and Anjela Oatridge. Non-linear registration with the variable viscosity fluid algorithm. In *Proceedings of Info Proc Med Imaging*, pages 238–251, Vancouver, September 1999.

[59] Yuying Li and Fadil Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Trans Image Proc*, 5(6):987–995, 1996.

[60] Nikos K. Logothetis, Jon Pauls, Mark Augath, Torsten Trinath, and Axel Oeltermann. Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412:150–157, 2001.

[61] L. C. Maas, B. D. Frederick, and P. F. Renshaw. Decoupled automated rotational and traslational registration for fMRI time series data: the DART registration algorithm. *Magn Reson Med*, 37:131–139, 1997.

[62] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Trans Med Imaging*, 16(2):187–198, 1997.

[63] Armando Manduca, Kiaran P. McGee, E. Brain Welch, Joel P. Felmlee, Roger Grimm, and Richard L. Ehman. Autocorrection in MR imaging: adaptive motion correction without navigator echoes. *Radiology*, 215(3):904–909, 2000.

[64] Armando Manduca, Kiaran P. McGee, E. Brian Welch, Joel P. Felmlee, and Richard L. Ehman. Fast automatic correction of motion artifacts in shoulder MRI. In *Proceedings of SPIE Med Imaging*, pages 853–859, San Diego, February 2001.

[65] Anne L. Martel, Alan R. Moody, Steven J. Allder, Gota S. Delay, and Paul S. Morgan. Extracting parametric images from dynamic contrast-enhanced MRI studies of the brain using factor analysis. *Med Image Anal*, 5:29–39, 2001.

[66] Klaus Mathiak and Stefan Posse. Evaluation of motion and realignment for functional magnetic resonance imaging in real time. *Magn Reson Med*, 45:167–171, 2001.

[67] Calvin R. Maurer, Jr., J. Michael Fitzpatrick, Matthew Y. Wang, Robert L. Galloway Jr., Robert J. Maciunas, and George S. Allen. Registration of head volume images using implantable fiducial markers. *IEEE Trans Med Imaging*, 16(4):447–462, 1997.

[68] Kiaran P. McGee, Joel P. Felmlee, Clifford R. Jack, Armando Manduca, and Richard L. Ehman. Estimation and correction of rotation induced artifacts using autocorrection and a linear distortion model. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 752, Glasgow, April 2001.

[69] Kiaran P. McGee, Joel P. Felmlee, Armando Manduca, Stephen J. Riederer, and Richard L. Ehman. Rapid autocorrection using prescan navigator echoes. *Magn Reson Med*, 43(4):583–588, 2000.

[70] V. L. Morgan, D. R. Pickens, S. L. Hartman, and R. R. Price. Comparison of functional MRI image realignment tools using a computer-generated phantom. *Magn Reson Med*, 46:510–514, 2001.

[71] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[72] Oscar Nestares and David J. Heeger. Robust multiresolution alignment of MRI brain volumes. *Magn Reson Med*, 43:705–715, 2000.

[73] Thanh D. Nguyen, Yi Wang, Richard Watts, and Ian Mitchell. k-space weighted least-squares algorithm for accurate and fast motion extraction from magnetic resonance navigator echoes. *Magn Reson Med*, 46:1037–1040, 2001.

[74] Christophoros Nikou, Fabrice Heitz, Jean-Paul Armspach, Izzie-Jacques Namer, and Daniel Grucker. Registration of MR/MR and MR/SPECT brain images by fast stochastic optimization of robust voxel similarity measures. *NeuroImage*, 8:30–43, 1998.

[75] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, second edition, 1999.

[76] Jeff Orchard and M. Stella Atkins. Theoretical analysis of the effect of fMRI brain activation on motion correction. In *Proceedings of the 10th Annual Meeting of ISMRM*, Honolulu, May 2002.

[77] R. J. Ordidge, J. A. Helpern, Z. X. Qing, R. A. Knight, and V. Nagesh. Correction of motional artifacts in diffusion-weighted MR images using navigator echoes. *Magn Reson Imag*, 12(3):455–460, 1994.

[78] John L. Ostuni, Attanagoda K. S. Santha, Venkata S. Mattay, Daniel R. Weinberger, Ronald L. Levin, and Joseph A. Frank. Analysis of interpolation effects in the reslicing of function MR images. *J Comput Assist Tomogr*, 21(5):803–810, 1997.

[79] James G. Pipe. Motion correction with PROPELLER MRI: Application to head motion and free-breathing cardiac imaging. *Magn Reson Med*, 42:963–969, 1999.

[80] James G. Pipe. Improved in-plane motion correction for PROPELLER MRI. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 743, Glasgow, April 2001.

[81] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1988.

[82] Davesh Raj, Derek Paley, Adam W. Anderson, Richard P. Kennan, and John C. Gore. Respiratory effects in functional magnetic resonance imaging due to bulk susceptibility changes. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 343, Glasgow, April 2001.

[83] N. F. Ramsey, J. S. van den Brink, M. M. C. van Muiswindle, P. J. M. Folkers, and C. T. W. Moonen. Phase navigator correction in 3D fMRI improves detection of brain activation: Qualitative assessment with a graded motor activation procedure. *NeuroImage*, 8:240–248, 1998.

[84] T. S. Sachs, Craig H. Meyer, Pablo Irarrazabal, Bob S. Hu, and Dwight G. Nishimura. The diminishing variance algorithm for real-time reduction of motion artifacts in MRI. *Magn Reson Med*, 34:412–422, 1995.

[85] Thomas Schmidt and Stephan B. Erberich. Optimized knowledge-based motion correction of fMRI time-series using parallel algorithms. In *Proceedings of SPIE Med Imaging*, pages 336–347, San Diego, February 2001.

[86] C. Studholme, D. L. G. Hill, and D. L. Hawkes. An overlap invariant entropy measure of 3D medical image alignent. *Pattern Recognition*, 32:71–86, 1999.

[87] S. Thesen, O. Heid, E. Mueller, and L. R. Schad. Prospective acquisition correction for head motion with image-based tracking for real-time fMRI. *Magn Reson Med*, 44:457–465, 2000.

[88] Chi-Ming Tsai and Dwight G. Nishimura. Reduced aliasing artifacts using variable-density k-space sampling trajectories. *Magn Reson Med*, 43:452–458, 2000.

[89] Marc Vailland and Christos Davatzikos. Hierarchical matching of cortical features for deformable brain image registration. In *Proceedings of Info Proc Med Imaging*, pages 182–195, Vancouver, September 1999.

[90] James T. Voyvodic. Real-time fMRI paradigm control, physiology, and behavior combined with near real-time statistical analysis. *NeuroImage*, 10:91–106, 1999.

[91] Y. Wang, R.C. Grimm, J.P. Felmlee, S.J. Riederer, and R.L. Ehman. Algorithms for extracting motion information from navigator echoes. *Magn Reson Med*, 36:117–123, 1996.

[92] B. Douglas Ward. *Deconvolution Analysis of FMRI Time Series Data*, 2001.

[93] H. Ward, S. Riederer, R. Grimm, R. Ehman, J. Felmlee, and C. Jack Jr. Prospective multiaxial motion correction for MRI. *Magn Reson Med*, 43:459–469, 2000.

[94] Heidi A. Ward, Armando Manduca, Stephen J. Riederer, and Clifford R. Jack Jr. Autocorrection for EPI ghost intensity reduction. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 295, Glasgow, April 2001.

[95] Edward Brian Welch and Armando Manduca. Motion correction from orthogonal k-space phase difference. In *Proceedings of the 9th Annual Meeting of ISMRM*, page 744, Glasgow, April 2001.

[96] Edward Brian Welch, Armando Manduca, Roger Grimm, Heidi Ward, and Clifford Jack Jr. Spherical navigator echoes for full 3-D rigid body motion measurement in MRI. In *Proceedings of SPIE Med Imaging*, pages 796–803, San Diego, February 2001.

[97] Edward Brian Welch, Armando Manduca, Roger C. Grimm, Heidi A. Ward, and Clifford R. Jack Jr. Spherical navigator echoes for full 3D rigid body motion measurement in MRI. *Magn Reson Med*, 47:32–41, 2002.

[98] Jay West, Michael Fitzpatrick, Matthew Y. Wang, Benoit M. Dewant, Calvin R. Maurer Jr., Robert M. Kessler, and Robert J. Maciunas. Retrospective intermodality registration techniques for images of the head: Surface-based versus volume-based. *IEEE Trans Med Imaging*, 18(2):144–150, 1999.

[99] M. L. Wood, M. J. Shivji, and P. L. Stanchev. Planar-motion correction with use of k-space data acquired in Fourier MR imaging. *J Magn Reson Imaging*, 5:57–64, 1995.

[100] R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid automated algorithm for aligning and reslicing PET images. *J Comput Assist Tomogr*, 16(4):620–633, 1992.

[101] R. P. Woods, S. T. Grafton, C. J. Holmes, S. R. Cherry, and J. C. Mazziotta. Automated image registration: I. General methods and intrasubject, intramodality validation. *J Comput Assist Tomogr*, 22:141–154, 1998.

[102] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. MRI-PET registration with automated algorithm. *J Comput Assist Tomogr*, 17(4):536–546, 1993.

[103] Roger P. Woods. Within-modality registration using intensity-based cost functions. In Isaac N. Bankman, editor, *Handbook of Medical Imaging: Processing and Analysis*, chapter 33, pages 529–536. Academic Press, 2000.

[104] B. Wowk, M. C. McIntyre, and J. K. Saunders. k-space detection and correction of physiological artifacts in fMRI. *Magn Reson Med*, 38(6):1029–1034, 1997.

[105] G. A. Write. Signal acquisition and processing for magnetic resonance imaging. In *Proceedings of Int Conf Image Proc*, pages 523–527, Austin, Texas, 1994.

[106] Mariangela Zamburlini, Vesna Sossi, R. De La Fuente-Fernandes, J. A. Stoessl, and T. J Ruth. Impact of different realignment algorithms on the SPM analysis of [11C]Raclopride PET studies. In *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference*, Norfolk, Virginia, November 2002.

[107] R. A. Zoroofi, K. Homma, Y. Sato, S. Tamura, and H. Naito. Technique for reduction of MRI 3-D affine motion artifacts. In *Proceedings of SPIE Med Imaging*, pages 1625–1634, San Diego, February 2000.