Chapter 2

Rigid Body Registration

John Ashburner & Karl J. Friston
The Wellcome Dept. of Imaging Neuroscience,
12 Queen Square, London WC1N 3BG, UK.

Contents

Abstract

Rigid body registration is one of the simplest forms of image registration, so this chapter provides an ideal framework for introducing some of the concepts that will be used by the more complex registration methods described later. The shape of a human brain changes very little with head movement, so rigid body transformations can be used to model different head positions of the same subject. Registration methods described in this chapter include within modality, or between different modalities such as PET and MRI. Matching of two images is performed by finding the rotations and translations that optimize some mutual function of the images. Within modality registration generally involves matching the images by minimizing the sum of squared difference between them. For between modality registration, the matching criterion needs to be more complex.

2.1 Introduction

Image registration is important in many aspects of functional image analysis. In imaging neuroscience, particularly for fMRI, the signal changes due to any hæmodynamic response can be small compared to apparent signal differences that can result from subject movement. Subject head movement in the scanner cannot be completely eliminated, so retrospective motion correction is performed as a preprocessing step. This is especially important for experiments where subjects may move in the scanner in a way that is correlated with the different conditions [12]. Even tiny systematic differences can result in a significant signal accumulating over numerous scans. Without suitable corrections, artifacts arising from subject movement correlated with the experimental paradigm may appear as activations. A second reason why motion correction is important is that it increases sensitivity. The t-test is based on the signal change relative to the residual variance. The residual variance is computed from the sum of squared differences between the data and the linear model to which it is fitted. Movement artifacts add to this residual variance, and so reduce the sensitivity of the test to true activations.

For studies of a single subject, sites of activation can be accurately localized by superimposing them on a high resolution structural image of the subject (typically a T1 weighted MRI). This requires registration of the functional images with the structural image. As in the case of movement correction, this is normally performed by optimizing a set of parameters describing a rigid body transformation, but the matching criterion needs to be more complex because the structural and functional images normally look very different. A further use for this registration is that a more precise spatial normalization can be achieved by computing it from a more detailed structural image. If the functional and structural images are in register, then a warp computed from the structural image can be applied to the functional images.

Another application of rigid registration is within the field of morphometry, and involves identifying shape changes within single subjects by subtracting coregistered images acquired at different times. The changes could arise for a number of different reasons, but most are related to pathology. Because the scans are of the same subject, the first step for this kind of analysis involves registering the images together by a rigid body transformation.

At its simplest, image registration involves estimating a mapping between a pair of images. One image is assumed to remain stationary (the reference image), whereas the other (the source image) is spatially transformed to match it. In order to transform the source to match the reference, it is necessary to determine a mapping from each voxel position in the reference to a corresponding position in the source. The source is then re-sampled at the new positions. The mapping can be thought of as a function of a set of estimated transformation parameters. A rigid-body transformation in three dimensions is defined by six parameters: three translations and three rotations.

There are two steps involved in registering a pair of images together. There is the *registration* itself, whereby the set of parameters describing a transformation is estimated. Then there is the *transformation*, where one of the images is transformed according to the estimated parameters. Performing the registration normally involves iteratively transforming the source image many times, using different parameters, until some matching criterion is optimized.

First of all, this chapter will explain how images are transformed via the process of re-sampling. This chapter is about about rigid registration of images, so the next section describes the parameterization of rigid-body transformations as a subset of the more general affine transformations. The final two sections describe methods of rigid-body registration, in both intra- and inter-modality contexts. Intra-modality registration implies registration of images acquired using the same modality and scanning sequence or contrast agent, whereas inter-modality registration allows the registration of different modalities (e.g., T1 to T2 weighted MRI, or MRI to PET).

2.2 Re-sampling Images

An image transformation is usually implemented as a "pulling" operation (where pixel values are pulled from the original image into their new location) rather than a "pushing" one (where the pixels in the original image are pushed into their new location). This involves determining for each voxel in the transformed image, the corresponding intensity in the original image. Usually, this requires sampling between the centers of voxels, so some form of interpolation is needed.

2.2.1 Simple Interpolation

The simplest approach is to take the value of the closest voxel to the desired sample point. This is referred to as *nearest neighbor* or *zero-order hold* re-sampling. This has the advantage that the original voxel intensities are preserved, but the resulting image is degraded quite considerably, resulting in the re-sampled image having a "blocky" appearance.

Another approach is to use *trilinear interpolation* (*first-order hold*) to re-sample the data. This is slower than nearest neighbor, but the resulting images are less "blocky". However, trilinear interpolation has the effect of losing some high frequency information from the image.

Figure 2.1 will now be used to illustrate bilinear interpolation (the two dimensionsional versio of trilinear interpolation). Assuming that there is a regular grid of pixels at co-ordinates x_a, y_a to x_p, y_p, having intensities v_a to v_p, and that the point to re-sample is at u. The value at points r and s are first determined (using linear interpolation) as follows:

$$\begin{aligned} v_r &= \frac{(x_g - x_r)v_f + (x_r - x_f)v_g}{x_g - x_f} \\ v_s &= \frac{(x_k - x_s)v_j + (x_s - x_j)v_k}{x_k - x_j} \end{aligned}$$

Then v_u is determined by interpolating between v_r and v_s:

$$v_u = \frac{(y_u - y_s)v_r + (y_r - y_u)v_s}{y_r - y_s}$$

The extension of the approach to three dimensions is trivial.

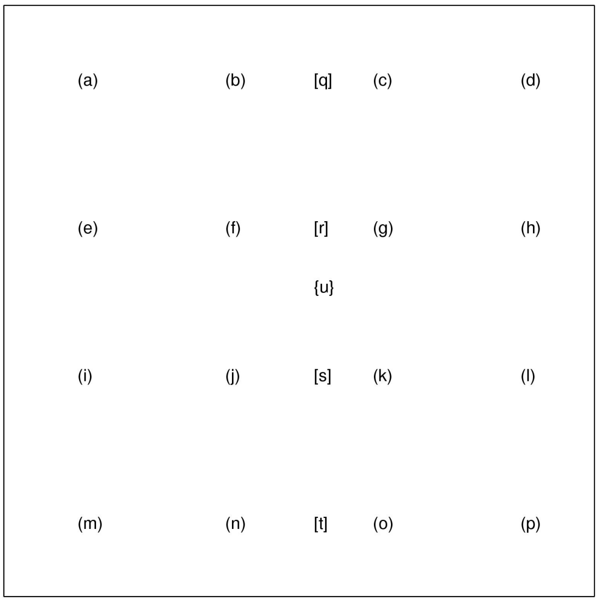

Figure 2.1: Illustration of image interpolation in two dimensions. Points a through to p represent the original regular grid of pixels. Point u is the point who's value is to be determined. Points q to t are used as intermediates in the computation.

2.2.2 Polynomial Interpolation

Rather than using only the 8 nearest neighbors (in 3D) to estimate the value at a point, more neighbors can be used in order to fit a smooth function through the neighboring voxels, and then read off the value of the function at the desired location. *Polynomial interpolation* is one such approach (zero- and first-order hold interpolations are simply low order polynomial interpolations). It is now illustrated how v_q can be determined from pixels a to d. The coefficients ($\mathbf{q}$) of a polynomial that runs through these points can be obtained by computing:

$$\mathbf{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & (x_b - x_a) & (x_b - x_a)^2 & (x_b - x_a)^3 \\ 1 & (x_c - x_a) & (x_c - x_a)^2 & (x_c - x_a)^3 \\ 1 & (x_d - x_a) & (x_d - x_a)^2 & (x_d - x_a)^3 \end{bmatrix}^{-1} \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \end{bmatrix}$$

Then v_q can be determined from these coefficients by:

$$v_q = \begin{bmatrix} 1 & (x_q - x_a) & (x_q - x_a)^2 & (x_q - x_a)^3 \end{bmatrix} \mathbf{q}$$

To determine v_u, a similar polynomial would be fitted through points q, r, s and t. The Vandermonde matrices required for polynomial interpolation are very ill conditioned, especially for higher orders. A better way of doing polynomial interpolation involves using *Lagrange polynomials* [see 22, 13]. Polynomial interpolation is a very crude approach, which has the disadvantage that discontinuities arise when moving from different sets of nearest neighbors.

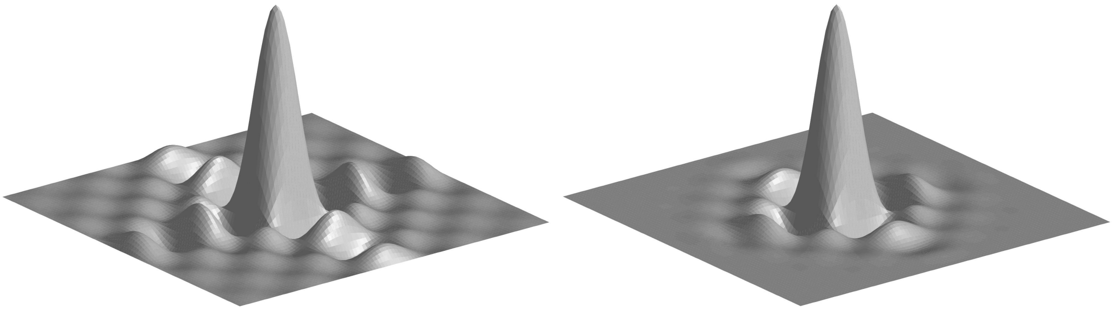

Figure 2.2: Sinc function in two dimensions, both with (right) and without (left) a Hanning window.

2.2.3 Windowed Sinc Interpolation

The optimum method of applying rigid-body transformations to images with minimal interpolation artifact is to do it in Fourier space. In real space, the interpolation method that gives results closest to a Fourier interpolation is *sinc* interpolation. This involves convolving the image with a sinc function centered on the point to be re-sampled. To perform a pure sinc interpolation, every voxel in the image should be used to sample a single point. This is not feasible due to speed considerations, so an approximation using a limited number of nearest neighbors is used. Because the sinc function extends to infinity, it is often truncated by modulating with a Hanning window (see Figure 2.2). Because the function is separable, the implementation of sinc interpolation is similar to that for polynomial interpolation, in that it is performed sequentially in the three dimensions of the volume. For one dimension the windowed sinc function using the I nearest neighbors would be:

$$\sum_{i=1}^{I} v_i \frac{\frac{sin(\pi d_i)}{\pi d_i}\frac{1}{2}\left(1+cos\left(2\pi d_i/I\right)\right)}{\sum_{j=1}^{I}\frac{sin(\pi d_j)}{\pi d_j}\frac{1}{2}\left(1+cos\left(2\pi d_j/I\right)\right)}$$

where d_i is the distance from the center of the ith voxel to the point to be sampled, and v_i is the value of the ith voxel.

2.2.4 Generalized Interpolation

The methods described so far are all classical interpolation methods that locally convolve the image with some form of interpolant[1]. Much more efficient re-sampling can be performed using *generalized interpolation* [27], where the images are first transformed into something else before applying the local convolution. Generalized interpolation methods model an image as a linear combination of basis functions with local support, typically *B-splines* or *o-Moms* (maximal-order interpolation of minimal support) basis functions (see Figure 2.3). Before re-sampling begins, an

[1] The polynomial interpolation can also be formulated this way by combining Equations 2.1 and 2.1 to eliminate the intermediate $\mathbf{q}$.

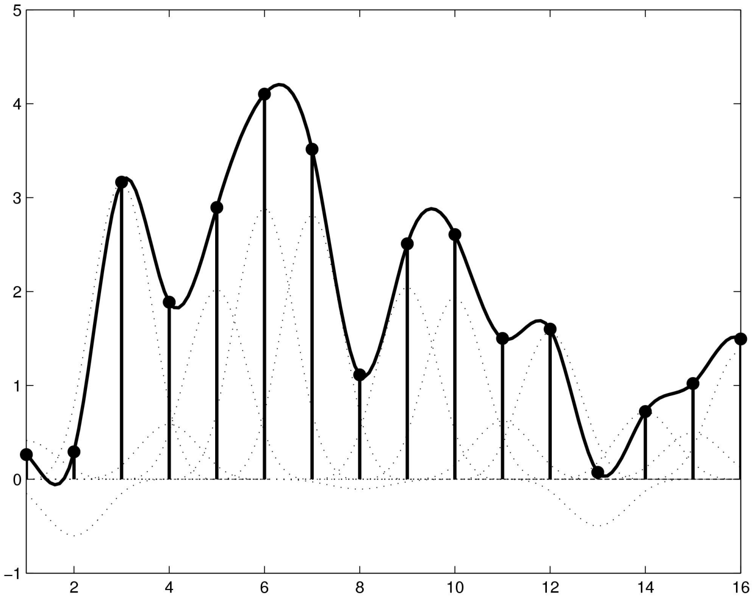

Figure 2.3: This figure illustrates a one dimensional B-spline representation of an image, where the image is assumed to be composed of a linear combination of B-spline basis functions. The dotted lines are the individual basis functions, which sum to produce the interpolated function (solid line).

image of basis function coefficients is produced, which involves a very fast deconvolution [29, 30]. Re-sampling at each new point then involves computing the appropriate linear combination of basis functions, which can be thought of as a local convolution of the basis function coefficients.

B-splines are a family of functions of varying degree. Interpolation using B-splines of degree 0 or 1 (first and second order) is identical to nearest neighbor[2] or linear interpolation respectively. B-splines of degree n are given by:

$$\beta^n(x) = \sum_{j=0}^{n} \frac{(-1)^j(n+1)}{(n+1-j)!j!} \max\left(\frac{n+1}{2} + x - j, 0\right)^n$$

An nth degree B-spline has a local support of $n+1$, which means that during the final re-sampling step, a linear combination of $n+1$ basis functions are needed to compute an interpolated value. o-Moms are derived from B-splines, and consist of a linear combination of the B-spline and its derivatives. They produce the most accurate interpolation for the least local support, but lack some of the B-splines' advantages. Unlike the o-Moms functions, a B-spline of order n is $n-1$ times continuously differentiable.

2.2.5 Fourier Methods

Higher order interpolation is slow when many neighboring voxels are used, but there are faster ways of interpolating when doing rigid-body transformations. Translations parallel to the axes are trivial, as these simply involve convolving with a translated delta function. For translations that are not whole numbers of pixels, the delta function is replaced by a sinc function centered

[2] Except with a slightly different treatment exactly in the center of two voxels.

at the translation distance. The use of fast Fourier transforms means that the convolution can be performed most rapidly as a multiplication in Fourier space. It is clear how translations can be performed in this way, but rotations are less obvious. One way that rotations can be effected involves replacing them by a series of shears [7] (see Section 2.3.4). A shear simply involves translating different rows or columns of an image by different amounts, so each shear can be performed as a series of one dimensional convolutions.

2.3 Rigid-body Transformations

Rigid-body transformations consist of only rotations and translations, and leave given arrangements unchanged. They are a subset of the more general affine[3] transformations. For each point (x_1, x_2, x_3) in an image, an affine mapping can be defined into the co-ordinates of another space (y_1, y_2, y_3). This is expressed as:

$$\begin{array}{rcllll} y_1 & = & m_{11}x_1 & + m_{12}x_2 & + m_{13}x_3 & + m_{14} \\ y_2 & = & m_{21}x_1 & + m_{22}x_2 & + m_{23}x_3 & + m_{24} \\ y_3 & = & m_{31}x_1 & + m_{32}x_2 & + m_{33}x_3 & + m_{34} \end{array}$$

which is often represented by a simple matrix multiplication ($\mathbf{y} = \mathbf{M}\mathbf{x}$):

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

The elegance of formulating these transformations in terms of matrices is that several of them can be combined, simply by multiplying the matrices together to form a single matrix. This means that repeated re-sampling of data can be avoided when re-orienting an image. Inverse affine transformations are obtained by inverting the transformation matrix.

2.3.1 Translations

If a point $\mathbf{x}$ is to be translated by $\mathbf{q}$ units, then the transformation is simply:

$$\mathbf{y} = \mathbf{x} + \mathbf{q}$$

In matrix terms, this can be considered as:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

2.3.2 Rotations

In two dimensions, a rotation is described by a single angle. Consider a point at co-ordinate (x_1, x_2) on a two dimensional plane. A rotation of this point to new co-ordinates (y_1, y_2), by θ radians around the origin, can be generated by the transformation:

$$\begin{array}{rcrcl} y_1 & = & cos(\theta)x_1 & + & sin(\theta)x_2 \\ y_2 & = & -sin(\theta)x_1 & + & cos(\theta)x_2 \end{array}$$

[3] Affine means that parallel lines remain parallel after the transformation.

This is another example of an affine transformation. For the three dimensional case, there are three orthogonal planes that an object can be rotated in. These planes of rotation are normally expressed as being around the axes. A rotation of q_1 radians about the first (x) axis is normally called pitch, and is performed by:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(q_1) & sin(q_1) & 0 \\ 0 & -sin(q_1) & cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

Similarly, rotations about the second (y) and third (z) axes (called roll and yaw respectively) are carried out by the following matrices:

$$\begin{bmatrix} cos(q_2) & 0 & sin(q_2) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(q_2) & 0 & cos(q_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} cos(q_3) & sin(q_3) & 0 & 0 \\ -sin(q_3) & cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Rotations are combined by multiplying these matrices together in the appropriate order. The order of the operations is important. For example, a rotation about the first axis of $\pi/2$ radians followed by an equivalent rotation about the second would produce a very different result to that obtained if the order of the operations was reversed.

2.3.3 Zooms

The affine transformations described so far will generate purely rigid-body mappings. Zooms are needed to change the size of an image, or to work with images whose voxel sizes are not isotropic, or differ between images. These represent scalings along the orthogonal axes, and can be represented via:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

A single zoom by a factor of -1 will flip an image (see Section 2.3.7). Two flips in different directions will merely rotate it by π radians (a rigid-body transformation). In fact, any affine transformation with a negative determinant will render the image flipped.

2.3.4 Shears

Shearing by parameters q_1, q_2 and q_3 can be performed by the following matrix:

$$\begin{bmatrix} 1 & q_1 & q_2 & 0 \\ 0 & 1 & q_3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A shear by itself is not a rigid-body transformation, but it is possible to combine shears in order to generate a rotation. In two dimensions, a matrix encoding a rotation of θ radians about the origin (see Section 2.4) can be constructed by multiplying together three matrices that effect shears [7]:

$$\begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & tan(\theta/2) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ sin(\theta) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & tan(\theta/2) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotations in three dimensions can be decomposed into four shears [6]. As shears can be performed quickly as one dimensional convolutions, then these decompositions are very useful for doing accurate and rapid rigid-body transformations of images.

2.3.5 Parameterizing a Rigid-body Transformation

When doing rigid registration of a pair of images, it is necessary to estimate six parameters that describe the rigid-body transformation matrix. There are many ways of parameterizing this transformation in terms of six parameters ($\mathbf{q}$). One possible form is:

$$\mathbf{M} = \mathbf{TR}$$

where:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(q_4) & sin(q_4) & 0 \\ 0 & -sin(q_4) & cos(q_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(q_5) & 0 & sin(q_5) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(q_5) & 0 & cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(q_6) & sin(q_6) & 0 & 0 \\ -sin(q_6) & cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sometimes it is desirable to extract transformation parameters from a matrix. Extracting these parameters $\mathbf{q}$ from $\mathbf{M}$ is relatively straightforward. Determining the translations is trivial, as they are simply contained in the fourth column of $\mathbf{M}$. This just leaves the rotations:

$$\mathbf{R} = \begin{bmatrix} c_5c_6 & c_5s_6 & s_5 & 0 \\ -s_4s_5c_6 - c_4s_6 & -s_4s_5s_6 + c_4c_6 & s_4c_5 & 0 \\ -c_4s_5c_6 + s_4s_6 & -c_4s_5s_6 - s_4c_6 & c_4c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where s_4, s_5 and s_6 are the sines, and c_4, c_5 and c_6 are the cosines of parameters q_4, q_5 and q_6 respectively. Therefore, provided that c_5 is not zero:

$$\begin{aligned} q_5 &= sin^{-1}(r_{13}) \\ q_4 &= atan2(r_{23}/cos(q_5), r_{33}/cos(q_5)) \\ q_6 &= atan2(r_{12}/cos(q_5), r_{11}/cos(q_5)) \end{aligned}$$

where $atan2$ is the four quadrant inverse tangent.

2.3.6 Working with Volumes of Differing or Anisotropic Voxel Sizes

Voxel sizes need be considered during image registration. Often, the images (say $\mathbf{f}$ and $\mathbf{g}$) will have voxels that are anisotropic. The dimensions of the voxels are also likely to differ between images of different modalities. For simplicity, a Euclidean space is used, where measures of distance are expressed in millimeters. Rather than transforming the images into volumes with cubic voxels that are the same size in all images, one can simply define affine transformation matrices that map from voxel co-ordinates into this Euclidean space. For example, if image $\mathbf{f}$ is of size $128 \times 128 \times 43$ and has voxels that are $2.1mm \times 2.1mm \times 2.45mm$, the following matrix can be defined:

$$\mathbf{M_f} = \begin{bmatrix} 2.1 & 0 & 0 & -135.45 \\ 0 & 2.1 & 0 & -135.45 \\ 0 & 0 & 2.45 & -53.9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This transformation matrix maps voxel co-ordinates to a Euclidean space whose axes are parallel to those of the image and distances are measured in millimeters, with the origin at the center of the image volume (i.e. $\mathbf{M_f} \begin{bmatrix} 64.5 & 64.5 & 22 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. A similar matrix can be defined for $\mathbf{g}$ ($\mathbf{M_g}$). Because modern MR image formats such as SPI (Standard Product Interconnect) generally contain information about image orientations in their headers, it is possible to extract this information to automatically compute values for $\mathbf{M_f}$ or $\mathbf{M_g}$. This makes it possible to easily register images together that were originally acquired in completely different orientations.

The objective of a rigid-body registration is to determine the affine transformation that maps the co-ordinates of image $\mathbf{g}$, to that of $\mathbf{f}$. To accomplish this, a rigid-body transformation matrix $\mathbf{M_r}$ is determined, such that $\mathbf{M_f}^{-1}\mathbf{M_r}^{-1}\mathbf{M_g}$ will map from voxels in $\mathbf{g}$ to those in $\mathbf{f}$. The inverse of this matrix maps from $\mathbf{f}$ to $\mathbf{g}$. Once $\mathbf{M_r}$ has been determined, $\mathbf{M_f}$ can be set to $\mathbf{M_r}\mathbf{M_f}$. From there onwards the mapping between the voxels of the two images can be achieved by $\mathbf{M_f}^{-1}\mathbf{M_g}$. Similarly, if another image ($\mathbf{h}$) is also registered with $\mathbf{g}$ in the same manner, then not only is there a mapping from $\mathbf{h}$ to $\mathbf{g}$ (via $\mathbf{M_g}^{-1}\mathbf{M_h}$), but there is also one from $\mathbf{h}$ to $\mathbf{f}$, which is simply $\mathbf{M_f}^{-1}\mathbf{M_h}$ (derived from $\mathbf{M_f}^{-1}\mathbf{M_g}\mathbf{M_g}^{-1}\mathbf{M_h}$).

2.3.7 Left- and Right-handed Co-ordinate Systems

Positions in space can be represented in either a left- or right-handed co-ordinate system (see Figure 2.4), where one system is a mirror image of the other. For example, the system used by the Talairach Atlas [26] is right-handed, because the first dimension (often referred to as the x direction) increases from left to right, the second dimension goes from posterior to anterior (back to front) and the third dimension increases from inferior to superior (bottom to top). The axes can be rotated by any angle, and they still retain their handedness. An affine transformation mapping between left and right-handed co-ordinate systems has a negative determinant, whereas one that maps between co-ordinate systems of the same kind will have a positive determinant. Because the left and right sides of a brain have similar appearances, care must be taken when re-orienting image volumes. Consistency of the co-ordinate systems can be achieved by performing any re-orientations using affine transformations, and checking the determinants of the matrices.

2.3.8 Rotating tensors

Diffusion tensor imaging (DTI) is becoming increasingly useful. These datasets are usually stored as six images containing a scalar field for each unique tensor element. It is worth noting that a rigid-body transformation of a DTI dataset is not a simple matter of rigidly rotating the

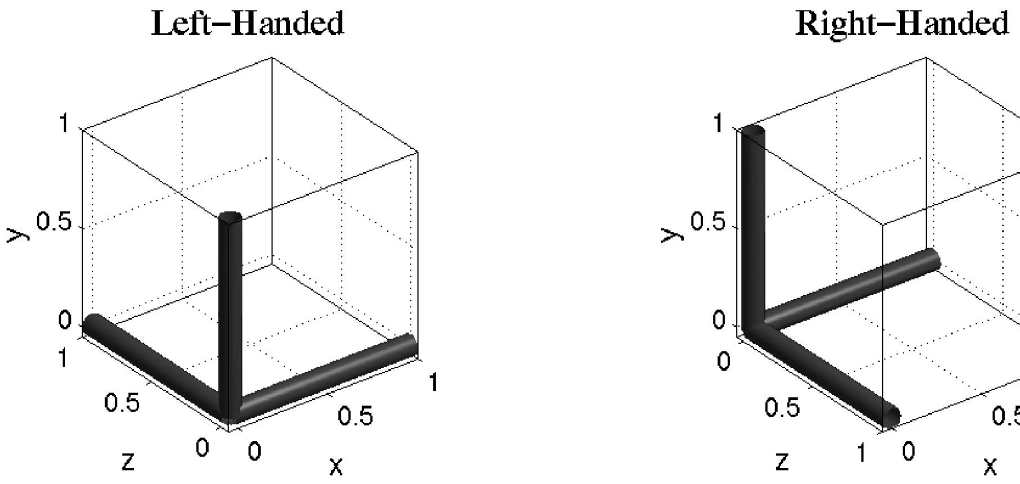

Figure 2.4: Left- and right-handed co-ordinate systems. The thumb corresponds to the x-axis, the index finger to the y-axis and the second finger to the z-axis.

individual scalar fields[4]. Once these fields have been re-sampled, the tensor represented at every voxel position needs to be rotated. A 3 × 3 tensor $\mathbf{T}$ can be rotated by a 3 × 3 matrix $\mathbf{R}$ by $\mathbf{T}' = \mathbf{R}\mathbf{T}\mathbf{R}^T$.

If DTI volumes are to be transformed using more complex warping models, then the local derivatives of the deformations (Jacobian matrices) need to be computed at each voxel. Suitable transformations can then be extracted from these derivatives, and applied to each element of the re-sampled tensor field [1, 2].

2.4 Within-Modality Rigid Registration

Whenever several images of the same subject have been acquired, it is extremely useful to have them all in register. Some of the simple benefits of this include allowing images to be averaged in order to increase signal to noise, or to subtract one image from another to emphasize differences between the images. Rigid[5] registration is normally used for retrospectively registering images of the same subject that have been collected at different times. Even if images were acquired during the same scanning session, the subject may have moved slightly between acquisitions.

The most common application of within-modality registration in functional imaging is to reduce motion artifacts by realigning the volumes in image time-series. The objective of realignment is to determine the rigid body transformations that best map the series of functional images to the same space. This can be achieved by minimizing the sum of squared differences between each of the images and a reference image, where the reference image could be one of the images in the series. For slightly better results, this procedure could be repeated, but instead of matching to one of the images from the series, the images would be registered to the mean of all the realigned images. Because of the non-stationary variance in the images, a variance image could be computed at the same time as the mean, in order to provide better weighting for the registration.

[4] It is worth noting that some interpolation methods are unsuitable for re-sampling the raw scalar fields, as the introduction of sampling errors can cause the positive definite nature of the tensors to be lost.

[5] Or affine registration if voxel sizes are not accurately known.

Voxels with a lot of variance should be given lower weighting, whereas those with less variance should be weighted more highly.

Within-modality image registration is also useful for looking at shape differences of brains. Morphometric studies sometimes involve looking at changes in brain shape over time, often to study the progression of a disease such as Alzheimers, or to monitor tumor growth or shrinkage. Differences between structural MR scans acquired at different times are identified, by first co-registering the images and then looking at the difference between the registered images. Rigid registration can also be used as a pre-processing step before using nonlinear registration methods for identifying shape changes [8].

Image registration involves estimating a set of parameters describing a spatial transformation that "best" match the images together. The goodness of the match is based on a *cost function*, which is maximized or minimized using some *optimization algorithm*. This section deals with registering images that have been collected using the same (or similar) modalities, allowing a relatively simple cost function to be used. In this case, the cost function is the mean squared difference between the images. Section 2.5 deals with the more complex task of registering images with different contrasts.

2.4.1 Optimization

The objective of optimization is to determine the values for a set of parameters for which some function of the parameters is minimized (or maximized). One of the simplest cases involves determining the optimum parameters for a model in order to minimize the sum of squared differences between a model and a set of real world data (χ^2). Normally there are many parameters, and it is not possible to exhaustively search through the whole parameter space. The usual approach is to make an initial parameter estimate, and begin iteratively searching from there. At each iteration, the model is evaluated using the current parameter estimates, and χ^2 computed. A judgement is then made about how the parameter estimates should be modified, before continuing on to the next iteration. The optimization is terminated when some convergence criterion is achieved (usually when χ^2 stops decreasing).

The image registration approach described here is essentially an optimization. One image (the source image) is spatially transformed so that it matches another (the reference image), by minimizing χ^2. The parameters that are optimized are those that describe the spatial transformation (although there are often other nuisance parameters required by the model, such as intensity scaling parameters). A good algorithm to use for rigid registration [10, 34] is *Gauss-Newton* optimization, and it is illustrated here.

Suppose that $b_i(\mathbf{q})$ is the function describing the difference between the source and reference images at voxel i, when the vector of model parameters have values $\mathbf{q}$. For each voxel, a first approximation of Taylor's Theorem can be used to estimate the value that this difference will take if the parameters $\mathbf{q}$ are decreased by $\mathbf{t}$:

$$b_i(\mathbf{q}-\mathbf{t}) \simeq b_i(\mathbf{q}) - t_1\frac{\partial b_i(\mathbf{q})}{\partial q_1} - t_2\frac{\partial b_i(\mathbf{q})}{\partial q_2}\dots$$

This allows the construction of a set of simultaneous equations (of the form $\mathbf{At} \simeq \mathbf{b}$) for estimating the values that $\mathbf{t}$ should assume to in order to minimize $\sum_i b_i(\mathbf{q}-\mathbf{t})^2$:

$$\begin{bmatrix} \frac{\partial b_1(\mathbf{q})}{\partial q_1} & \frac{\partial b_1(\mathbf{q})}{\partial q_2} & \dots \\ \frac{\partial b_2(\mathbf{q})}{\partial q_1} & \frac{\partial b_2(\mathbf{q})}{\partial q_2} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \end{bmatrix} \simeq \begin{bmatrix} b_1(\mathbf{q}) \\ b_2(\mathbf{q}) \\ \vdots \end{bmatrix}$$

From this, an iterative scheme can be derived for improving the parameter estimates. For iteration n, the parameters $\mathbf{q}$ are updated as:

$$\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} - \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \qquad (2.1)$$

where $\mathbf{A} = \begin{bmatrix} \frac{\partial b_1(\mathbf{q})}{\partial q_1} & \frac{\partial b_1(\mathbf{q})}{\partial q_2} & \cdots \\ \frac{\partial b_2(\mathbf{q})}{\partial q_1} & \frac{\partial b_2(\mathbf{q})}{\partial q_2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} b_1(\mathbf{q}) \\ b_2(\mathbf{q}) \\ \vdots \end{bmatrix}$.

This process is repeated until χ^2 can no longer be decreased - or for a fixed number of iterations. There is no guarantee that the best global solution will be reached, because the algorithm can get caught in a local minimum. To reduce this problem, the starting estimates for $\mathbf{q}$ should be set as close as possible to the optimum solution. The number of potential local minima can also be decreased by working with smooth images. This also has the effect of making the first order Taylor approximation more accurate for larger displacements. Once the registration is close to the final solution, it can continue with less smooth images.

In practice, $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ from Eqn. 2.1 are often computed 'on the fly' for each iteration. By computing these matrices using only a few rows of $\mathbf{A}$ and $\mathbf{b}$ at a time, much less computer memory is required than is necessary for storing the whole of matrix $\mathbf{A}$. Also, the partial derivatives $\partial b_i(\mathbf{q})/\partial q_j$ can be rapidly computed from the gradients of the images using the chain rule (see [32] for detailed information).

It should be noted that element i of $\mathbf{A}^T\mathbf{b}$ is equal to $\frac{1}{2}\frac{\partial \chi^2}{\partial q_i}$, and that element i,j of $\mathbf{A}^T\mathbf{A}$ is approximately equal to $\frac{1}{2}\frac{\partial^2\chi^2}{\partial q_i \partial q_j}$ (one half of the Hessian matrix, often referred to as the curvature matrix - see [22], Section 15.5 for a general description, or [32, 33] for more information related to image registration). Another way of thinking about the optimization is that it fits a quadratic function to the error surface at each iteration. Successive parameter estimates are chosen such that they are at the minimum point of this quadratic (illustrated for a single parameter in Figure 2.5).

2.4.2 Implementation

This section is about estimating parameters that describe a rigid-body transformation, but the principles can be extended to models that describe nonlinear warps. To register a source image $\mathbf{f}$ to a reference image $\mathbf{g}$, a six parameter rigid-body transformation (parameterized by q_1 to q_6) would be used. To perform the registration, a number of points in the reference image (each denoted by $\mathbf{x}_i$) are compared with points in the source image (denoted by $\mathbf{Mx}_i$, where $\mathbf{M}$ is the rigid-body transformation matrix constructed from the six parameters). The images may be scaled differently, so an additional intensity scaling parameter (q_7) may be included in the model. The parameters ($\mathbf{q}$) are optimized by minimizing the sum of squared differences[6] between the images according to the algorithm described in Sections 2.3.5 and 2.4.1 (Eqn. 2.1). The function that is minimized is:

$$\sum_i \left(f(\mathbf{Mx}_i) - q_7 g(\mathbf{x}_i)\right)^2$$

[6] Strictly speaking, it is the mean squared difference that is minimized, rather than the sum of squared differences. Inevitably, some values of $\mathbf{Mx}_i$ will lie outside the domain of $\mathbf{f}$, so nothing is known about what the image intensity should be at these points. The computations are only performed for points where both $\mathbf{x}_i$ and $\mathbf{Mx}_i$ lie within the field of view of the images.

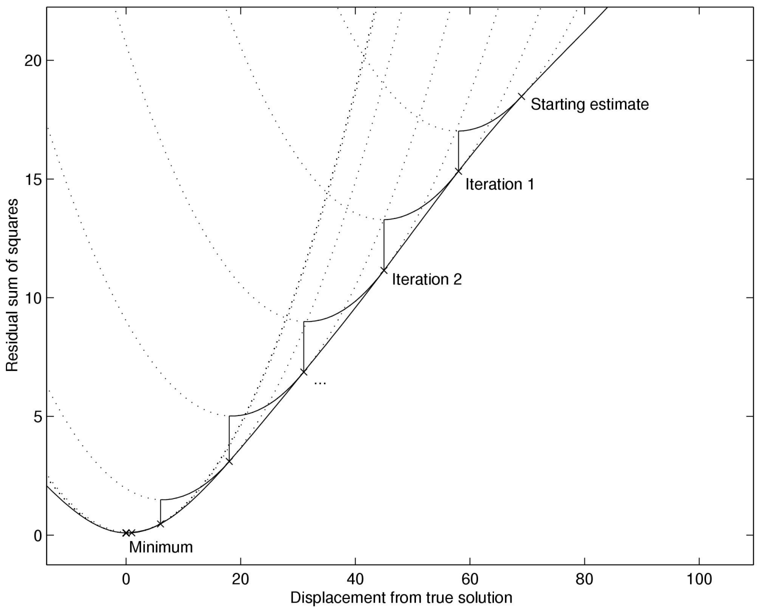

Figure 2.5: The optimization can be thought of as fitting a series of quadratics to the error surface. Each parameter update is such that it falls at the minimum of the quadratic.

where $\mathbf{M} = \mathbf{M_f}^{-1}\mathbf{M_r}^{-1}\mathbf{M_g}$, and $\mathbf{M_r}$ is constructed from parameters $\mathbf{q}$ (refer to Section 2.3.6). Vector $\mathbf{b}$ is generated for each iteration as:

$$\mathbf{b} = \begin{bmatrix} f(\mathbf{Mx}_1) - q_7 g(\mathbf{x}_1) \\ f(\mathbf{Mx}_2) - q_7 g(\mathbf{x}_2) \\ \vdots \end{bmatrix}$$

Each column of matrix $\mathbf{A}$ is constructed by differentiating $\mathbf{b}$ with respect to parameters q_1 to q_7:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f(\mathbf{Mx}_1)}{\partial q_1} & \frac{\partial f(\mathbf{Mx}_1)}{\partial q_2} & \cdots & \frac{\partial f(\mathbf{Mx}_1)}{\partial q_6} & -g(\mathbf{x}_1) \\ \frac{\partial f(\mathbf{Mx}_2)}{\partial q_1} & \frac{\partial f(\mathbf{Mx}_2)}{\partial q_2} & \cdots & \frac{\partial f(\mathbf{Mx}_2)}{\partial q_6} & -g(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{bmatrix}$$

Because non-singular affine transformations are easily invertible, it is possible to make the registration more robust by also considering what happens with the inverse transformation. By swapping around the source and reference image, the registration problem also becomes one of minimizing:

$$\sum_j (g(\mathbf{M}^{-1}\mathbf{y}_j) - q_7^{-1} f(\mathbf{y}_j))^2$$

In theory, a more robust solution could be achieved by simultaneously including the inverse transformation to make the registration problem symmetric [34]. The cost function would then be:

$$\lambda_1 \sum_i (f(\mathbf{Mx}_i) - q_7 g(\mathbf{x}_i))^2 + \lambda_2 \sum_j (g(\mathbf{M}^{-1}\mathbf{y}_j) - q_7^{-1} f(\mathbf{y}_j))^2$$

Normally, the intensity scaling of the image pair will be similar, so equal values for the weighting factors (λ_1 and λ_2) can be used. Matrix $\mathbf{A}$ and vector $\mathbf{b}$ would then be formulated as:

$$\mathbf{b} = \begin{bmatrix} \lambda_1^{\frac{1}{2}}(f(\mathbf{Mx}_1) - q_7 g(\mathbf{x}_1)) \\ \lambda_1^{\frac{1}{2}}(f(\mathbf{Mx}_2) - q_7 g(\mathbf{x}_2)) \\ \vdots \\ \lambda_2^{\frac{1}{2}}(g(\mathbf{M}^{-1}\mathbf{y}_1) - q_7^{-1} f(\mathbf{y}_1)) \\ \lambda_2^{\frac{1}{2}}(g(\mathbf{M}^{-1}\mathbf{y}_2) - q_7^{-1} f(\mathbf{y}_2)) \\ \vdots \end{bmatrix}$$

and

$$\mathbf{A} = \begin{bmatrix} \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{Mx}_1)}{\partial q_1} & \ldots & \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{Mx}_1)}{\partial q_6} & -\lambda_1^{\frac{1}{2}} g(\mathbf{x}_1) \\ \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{Mx}_2)}{\partial q_1} & \ldots & \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{Mx}_2)}{\partial q_6} & -\lambda_1^{\frac{1}{2}} g(\mathbf{x}_2) \\ \vdots & \ddots & \vdots & \vdots \\ \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_1)}{\partial q_1} & \ldots & \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_1)}{\partial q_6} & \lambda_2^{\frac{1}{2}} q_7^{-2} f(\mathbf{y}_1) \\ \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_2)}{\partial q_1} & \ldots & \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_2)}{\partial q_6} & \lambda_2^{\frac{1}{2}} q_7^{-2} f(\mathbf{y}_2) \\ \vdots & \ddots & \vdots & \vdots \end{bmatrix}$$

2.4.3 Residual Artifacts from PET and fMRI

Even after realignment, there may still be some motion related artifacts remaining in functional data. After retrospective realignment of PET images with large movements, the primary source of error is due to incorrect attenuation correction. In emission tomography methods, many photons are not detected because they are attenuated by the subject's head. Normally, a transmission scan (using a moving radioactive source external to the subject) is acquired before collecting the emission scans. The ratio of the number of detected photon pairs from the source, with and without a head in the field of view, produces a map of the proportion of photons that are absorbed along any line-of-response. If a subject moves between the transmission and emission scans, then the applied attenuation correction is incorrect because the emission scan is no longer aligned with the transmission scan. There are methods for correcting these errors [4], but they are beyond the scope of this book.

In fMRI, there are many sources of motion related artifacts. The most obvious ones are:

- Interpolation error from the re-sampling algorithm used to transform the images can be one of the main sources of motion related artifacts. When the image series is re-sampled, it is important to use a very accurate interpolation method.
- When MR images are reconstructed, the final images are usually the modulus of the initially complex data, resulting in any voxels that should be negative being rendered positive. This has implications when the images are re-sampled, because it leads to errors at the edge of the brain that cannot be corrected however good the interpolation method is. Possible ways to circumvent this problem are to work with complex data, or possibly to apply a low pass filter to the complex data before taking the modulus.
- The sensitivity (slice selection) profile of each slice also plays a role in introducing artifacts [20].

- fMRI images are spatially distorted, and the amount of distortion depends partly upon the position of the subject's head within the magnetic field. Relatively large subject movements result in the brain images changing shape, and these shape changes cannot be corrected by a rigid body transformation [15, 3].
- Each fMRI volume of a series is currently acquired a plane at a time over a period of a few seconds. Subject movement between acquiring the first and last plane of any volume is another reason why the images may not strictly obey the rules of rigid body motion.
- After a slice is magnetized, the excited tissue takes time to recover to its original state, and the amount of recovery that has taken place will influence the intensity of the tissue in the image. Out of plane movement will result in a slightly different part of the brain being excited during each repeat. This means that the spin excitation will vary in a way that is related to head motion, and so leads to more movement related artifacts [11].
- Nyquist ghost artifacts in MR images do not obey the same rigid body rules as the head, so a rigid rotation to align the head will not mean that the ghosts are aligned. The same also applies to other image artifacts such as those arising due to chemical shifts.
- The accuracy of the estimated registration parameters is normally in the region of tens of μm. This is dependent upon many factors, including the effects just mentioned. Even the signal changes elicited by the experiment can have a slight effect (a few μm) on the estimated parameters [9].

These problems cannot be corrected by simple image realignment, and so may be sources of possible stimulus correlated motion artifacts. Systematic movement artifacts resulting in a signal change of only one or two percent can lead to highly significant false positives over an experiment with many scans. This is especially important for experiments where some conditions may cause slight head movements (such as motor tasks, or speech), because these movements are likely to be highly correlated with the experimental design. In cases like this, it is difficult to separate true activations from stimulus correlated motion artifacts. Providing there are enough images in the series and the movements are small, some of these artifacts can be removed by using an ANCOVA model to remove any signal that is correlated with functions of the movement parameters [11]. However, when the estimates of the movement parameters are related to the the experimental design, it is likely that much of the true fMRI signal will also be lost. These are still unresolved problems.

2.5 Between-Modality Rigid Registration

The combination of multiple imaging modalities can provide enhanced information that is not readily apparent on inspection of individual image modalities. For studies of a single subject, sites of activation can be accurately localized by superimposing them on a high resolution structural image of the subject (typically a T1 weighted MRI). This requires registration of the functional images with the structural image. A further possible use for this registration is that a more precise spatial normalization can be achieved by computing it from a more detailed structural image. If the functional and structural images are in register, then a warp computed from the structural image can be applied to the functional images. Normally a rigid-body model is used for registering images of the same subject, but because fMRI images are usually severely distorted – particularly in the phase encode direction [15, 14] – it is often preferable to do non-linear registration [23, 16]. Rigid registration models require voxel sizes to be accurately known. This is a problem that is particularly apparent when registering images from different scanners.

Two images from the same subject acquired using the same modality or scanning sequences generally look similar, so it suffices to find the rigid-body transformation parameters that minimize the sum of squared differences between them. However, for co-registration between modalities there is nothing quite as obvious to minimize, as there is no linear relationship between the image intensities (see Figure 2.9).

Older methods of registration involved the manual identification of homologous landmarks in the images. These landmarks are aligned together, thus bringing the images into registration. This is time-consuming, requires a degree of experience, and can be rather subjective. One of the first widely used semi-automatic co-registration methods was that known as the "head-hat" approach [21]. This method involved extracting brain surfaces of the two images, and then matching the surfaces together. There are also a number of other between-modality registration methods that involve partitioning the images, of finding common features between them, and then registering them together, but they are beyond the scope of this chapter.

The first intensity based inter-modal registration method was *AIR* [35], which has been widely used for a number of years for registering PET and MR images. This method uses a variance of intensity ratios (VIR) cost function, and involves dividing the MR images into a number of partitions based on intensity. The registration is approximately based on minimizing the variance of the corresponding PET voxel intensities for each partition. It makes a number of assumptions about how the PET intensity varies with the MRI intensity, which are generally valid within the brain, but do not work when non-brain tissue is included. Because of this, the method has the disadvantage of requiring the MR images to be pre-processed, which normally involves editing to remove non-brain tissue. For a review of a number of inter-modality registration approaches up until the mid 1990s, see [36].

2.5.1 Information Theoretic Approaches

The most recent voxel-similarity measures to be used for inter-modal (as well as intra-modal [18]) registration have been based on *information theory*. These measures are based on joint probability distributions of intensities in the images, usually discretely represented in the form of 2D joint histograms. Once constructed, the joint histogram is normalized so that the bins integrate to unity.

The first information theoretic measure to be proposed was the entropy of the joint probability distribution [24], which should be minimized when the images are in register:

$$H(\mathbf{f},\mathbf{g}) \quad = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(\mathbf{f},\mathbf{g}) \log P(\mathbf{f},\mathbf{g}) d\mathbf{f} d\mathbf{g}$$

The discrete representation of the probability distributions is from a joint histogram (that has been normalized to sum to unity), which can be considered as an I by J matrix $\mathbf{P}$. The entropy is then computed from the histogram according to:

$$H(\mathbf{f},\mathbf{g}) \quad = \sum_{j=1}^{J}\sum_{i=1}^{I} p_{ij} \log p_{ij}$$

In practice, the entropy measure was found to produce poor registration results, but shortly afterwards, a more robust measure of registration quality was introduced. This was based on *mutual information* (MI) [5, 31] (also known as *Shannon information*), which is given by:

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

where $H(\mathbf{f},\mathbf{g})$ is the joint entropy of the images, and $H(\mathbf{f})$ and $H(\mathbf{g})$ are their marginalized entropies given by:

$$\begin{aligned} H(\mathbf{f}) &= -\textstyle\int_{-\infty}^{\infty} P(\mathbf{f}) \log P(\mathbf{f}) d\mathbf{f} \\ H(\mathbf{g}) &= -\textstyle\int_{-\infty}^{\infty} P(\mathbf{g}) \log P(\mathbf{g}) d\mathbf{g} \end{aligned}$$

MI is a measure of dependence of one image on the other, and can be considered as the distance (Kullback-Leibler divergence) between the joint distribution ($P(\mathbf{f},\mathbf{g})$) and the distribution assuming complete independence ($P(\mathbf{f})P(\mathbf{g})$). When the two distributions are identical, this distance (and the mutual information) is zero. After rearranging, the expression for MI becomes:

$$I(\mathbf{f},\mathbf{g}) = KL(P(\mathbf{f},\mathbf{g})||P(\mathbf{f})P(\mathbf{g})) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(\mathbf{f},\mathbf{g}) \log\left(\frac{P(\mathbf{f},\mathbf{g})}{P(\mathbf{f})P(\mathbf{g})}\right) d\mathbf{f}d\mathbf{g}$$

It is assumed that the MI between the images is maximized when they are in register (see Figure 2.6). One problem though, is that MI is biased by the amount of overlap between the images (although it is still less influenced than the joint entropy). When there is less overlap, fewer samples are used to construct a joint histogram, meaning that it is more "spiky". This produces a slightly higher measure of MI. One overlap invariant information theoretic measure [25] that can be used for registration is:

$$\tilde{I}(\mathbf{f},\mathbf{g}) = \frac{H(\mathbf{f}) + H(\mathbf{g})}{H(\mathbf{f},\mathbf{g})}$$

Another useful measure [19] is:

$$\tilde{I}(\mathbf{f},\mathbf{g}) = 2H(\mathbf{f},\mathbf{g}) - H(\mathbf{f}) - H(\mathbf{g})$$

and also the *entropy correlation coefficient* [19] [see 22, page 634 for more information]:

$$U(\mathbf{f},\mathbf{g}) = 2\frac{H(\mathbf{f}) + H(\mathbf{g}) - H(\mathbf{f},\mathbf{g})}{H(\mathbf{f}) + H(\mathbf{g})}$$

2.5.2 Implementation Details

Generating a joint histogram involves scanning through the voxels of the reference image and finding the corresponding points of the source. The appropriate bin in the histogram is incremented by one for each of these point pairs. Pairs are ignored if the corresponding voxel is unavailable because it lies outside the image volume. The co-ordinate of the corresponding point rarely lies at an actual voxel center, meaning that interpolation is required.

Many developers use *partial volume interpolation* [5], rather than interpolating the images themselves, but this can make the MI cost function particularly susceptible to interpolation artifact (see Figure 2.8). The MI tends to be higher when voxel centers are sampled, where one is added to a single histogram bin. MI is lower when sampling in the center of the eight neighbors, as an eighth is added to eight bins. These artifacts are especially prominent when fewer point pairs are used to generate the histograms.

A simpler alternative is to interpolate the images themselves, but this can lead to new intensity values in the histograms, which also cause interpolation artifacts. This artifact largely occurs because of aliasing after integer represented images are rescaled so that they have values between

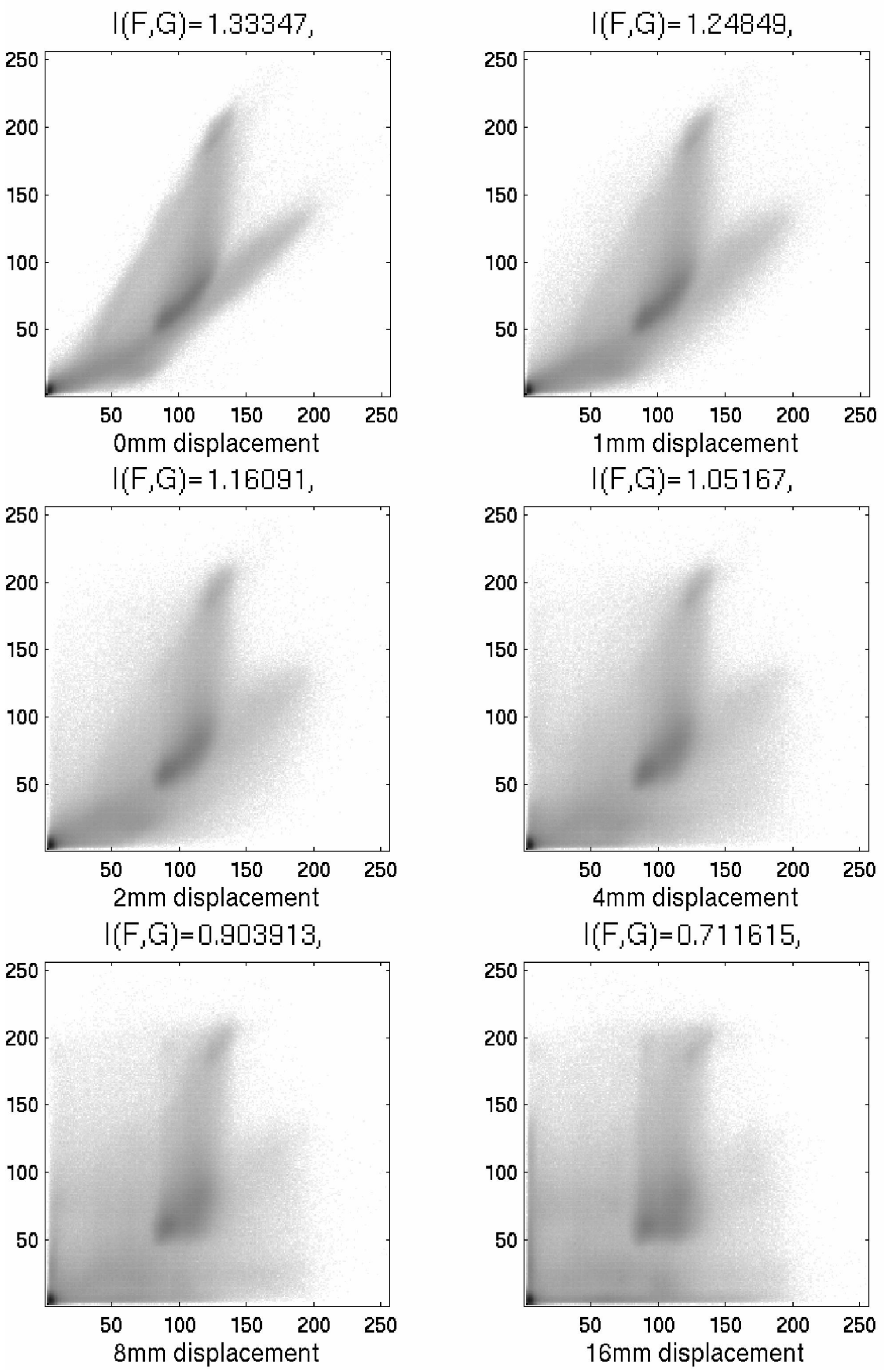

Figure 2.6: An illustration of how the joint histogram of an image pair changes as they are displaced relative to each other (note that the pictures show $\log(1 + N)$, where N is the count in each histogram bin). The MI of the images is also shown.

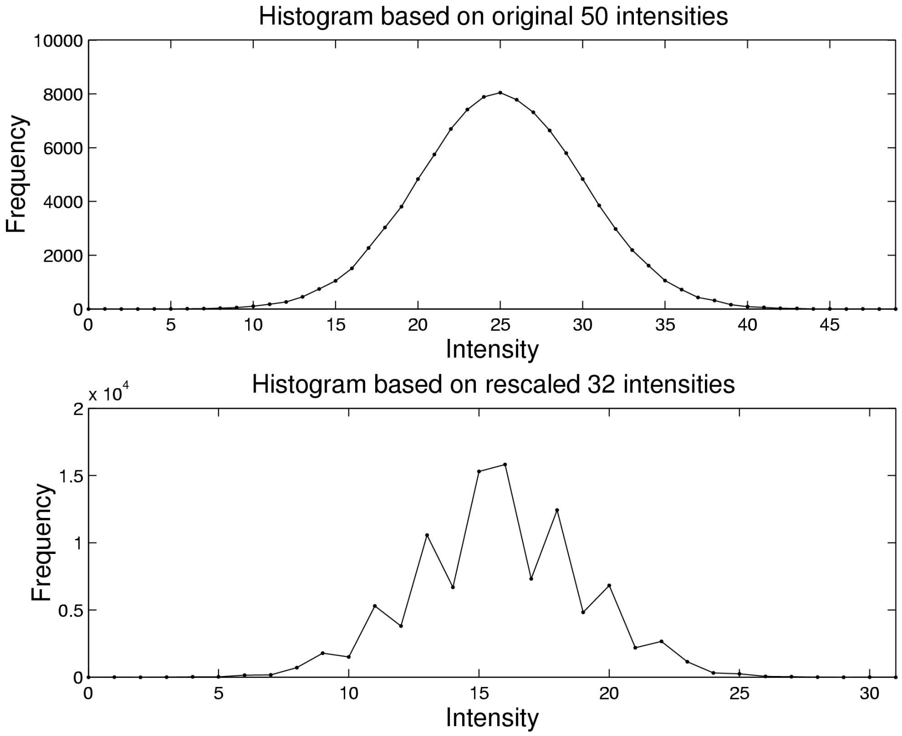

Figure 2.7: Rescaling an image can lead to aliasing artifacts in its histogram. Above: histogram based on original integer intensity values, simulated to have a Gaussian distribution. Below: the histogram after the intensities are rescaled to between zero and 63 shows aliasing artifacts.

zero and $I-1$, where I is the number of bins in the histogram (see Figure 2.7). If care is taken at this stage, then interpolation of the image intensities becomes less of a problem. Another method of reducing these artifacts is to not sample the reference image on a regular grid, by (for example) introducing a random jitter to the sampled points [17].

Histograms contain noise, especially if a relatively small number of points are sampled in order to generate them. The optimum binning to use is still not fully resolved, and is likely to vary from application to application, but most researchers use histograms ranging between about 16 × 16 and 256 × 256. Smoothing a histogram has a similar effect to using fewer bins. Another alternative is to use a continuous representation of the joint probability distribution, such as a Parzen window density estimate [31], or possibly even a Gaussian mixture model representation.

Section 2.4.1 introduced a method of optimization based on the first and second derivatives of the cost function. Similar principles have been be applied to minimizing the VIR cost function [35], and also to maximizing MI [28][7]. However, the most widely adopted scheme for maximizing MI is Powell's Method [see 22, , page 415], which involves a series of successive line searches. Failures occasionally arise if the voxel similarity measure does not vary smoothly with changes to the parameter estimates. This can happen because of interpolation artifact, or if insufficient data contributes to the joint histogram. Alternatively, the algorithm can get caught within a local optimum, so it is important to assign starting estimates that approximately register the images. The required accuracy of the starting estimates depends on the particular images, but

[7] This paper uses Levenberg-Marquardt optimization [22], which is a stabilized version of the Gauss-Newton method

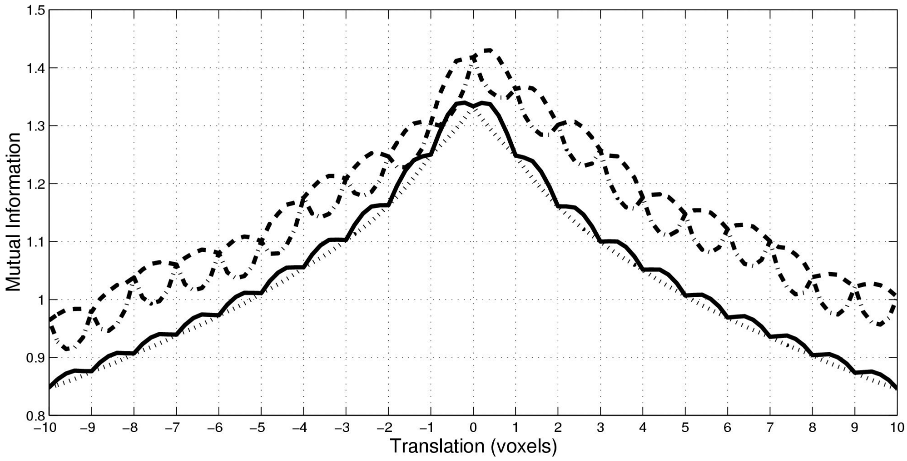

Figure 2.8: The mutual information cost function can be particularly susceptible to interpolation artifacts. This figure shows a plot of the MI between two images when they are translated with respect to each other. The dotted and dot-dashed lines show it computed using partial volume interpolation at high and lower sampling densities. The solid and dashed lines show MI computed by interpolating the images themselves.

an approximate figure for many brain images with a good field of view would be in the region of about 5cm for translations and 15^o for rotations.

Bibliography

[1] D. C. Alexander, J. C. Gee, and R. Bajcsy. Strategies for data reorientation during non-rigid transformations of diffusion tensor images,. *Lecture Notes in Computer Science*, 1679:463–472, 1999.

[2] D. C. Alexander, C. Pierpaoli, P. J. Basser, and J. C. Gee. Spatial transformations of diffusion tensor magnetic resonance images. *IEEE Transactions on Medical Imaging*, 20(11):1131–1139, 2001.

[3] J. L. R. Andersson, C. Hutton, J. Ashburner, R. Turner, and K. J. Friston. Modeling geometric deformations in epi time series. *NeuroImage*, 13(5):903–919, 2001.

[4] J. L. R. Andersson, B. E. Vagnhammar, and H. Schneider. Accurate attenuation correction despite movement during PET imaging. *Journal of Nuclear Medicine*, 36(4):670–678, 1995.

[5] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Proc. Information Processing in Medical Imaging*, pages 263–274, Dordrecht, The Netherlands, 1995. Kluwer Academic Publishers.

[6] R. W. Cox and A. Jesmanowicz. Real-time 3D image registration for functional MRI. *Magnetic Resonance In Medicine*, 42:1014–1018, 1999.

[7] W. F. Eddy, M. Fitzgerald, and D. C. Noll. Improved image registration by using Fourier interpolation. *Magnetic Resonance in Medicine*, 36:923–931, 1996.

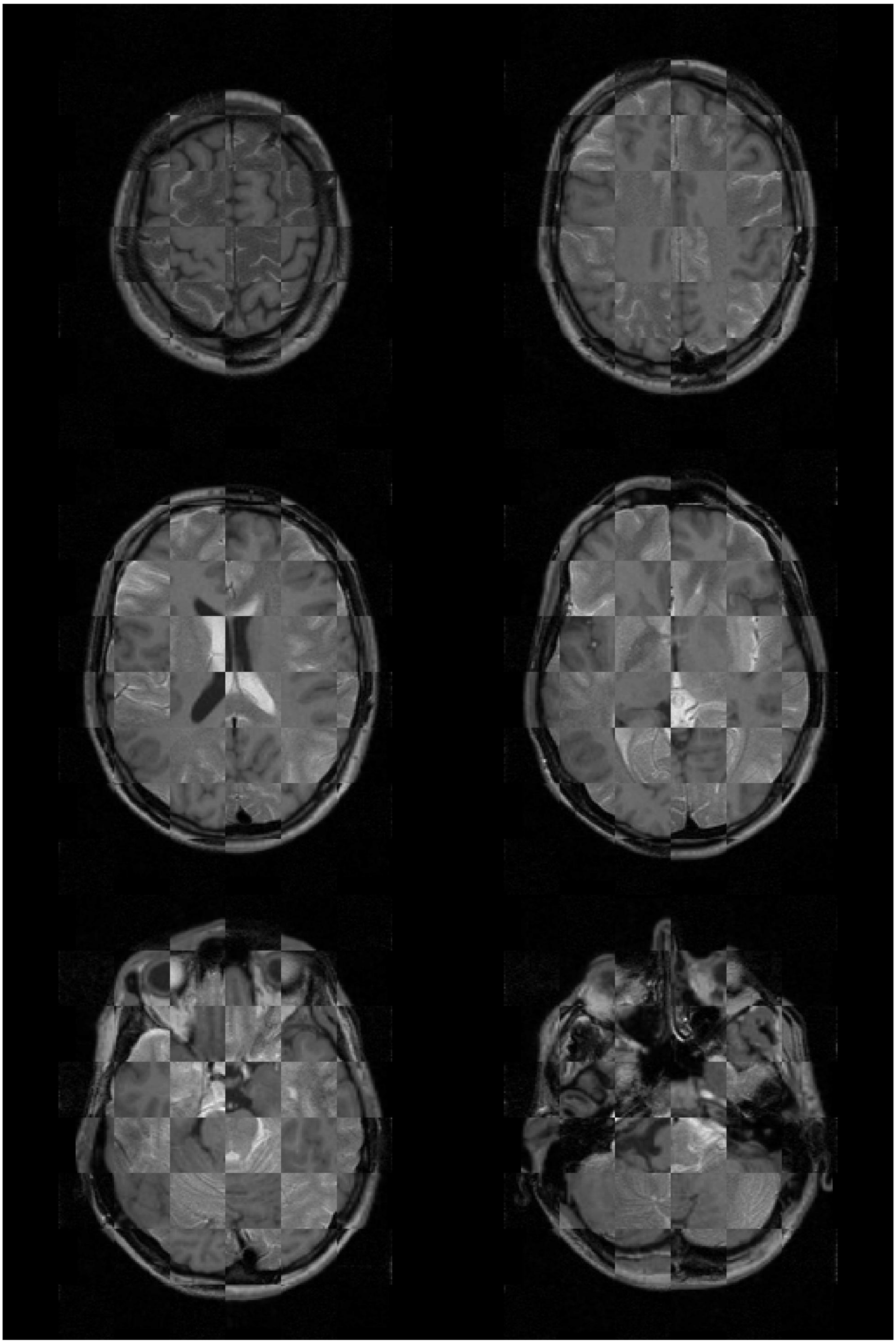

Figure 2.9: An example of T1 and T2 weighted MR images registered using mutual information. The two registered images are shown interleaved in a chequered pattern.

[8] P. A. Freeborough and N. C. Fox. Modelling brain deformations in alzheimer disease by fluid registration of serial MR images. *Journal of Computer Assisted Tomography*, 22(5):838–843, 1998.

[9] L. Freire and J.-F. Mangin. Motion correction algorithms of the brain mapping community create spurious functional activations. In M. F. Insana and R. M. Leah, editors, *Proc. Information Processing in Medical Imaging*, pages 246–258, Berlin, Heidelberg, 2001. Springer-Verlag.

[10] K. J. Friston, J. Ashburner, C. D. Frith, J.-B. Poline, J. D. Heather, and R. S. J. Frackowiak. Spatial registration and normalization of images. *Human Brain Mapping*, 2:165–189, 1995.

[11] K. J. Friston, S. Williams, R. Howard, R. S. J. Frackowiak, and R. Turner. Movement-related effects in fMRI time-series. *Magnetic Resonance in Medicine*, 35:346–355, 1996.

[12] J. V. Hajnal, R. Mayers, A. Oatridge, J. E. Schwieso, J. R. Young, and G. M. Bydder. Artifacts due to stimulus correlated motion in functional imaging of the brain. *Magnetic Resonance in Medicine*, 31:289–291, 1994.

[13] A. K. Jain. *Fundamentals of Digital Image Processing.* Pretence-Hall, 1989.

[14] P. Jezzard. *Handbook of Medical Imaging*, chapter 26, pages 425–438. Academic Press, 2000.

[15] P. Jezzard and S. Clare. Source distortion in functional MRI data. *Human Brain Mapping*, 8(2):80–85, 1999.

[16] J. Kybic, P. Thévenaz, A. Nirkko, and M. Unser. Unwarping of unidirectionally distorted EPI images. *IEEE Transactions on Medical Imaging*, 19(2):80–93, 2000.

[17] B. Likar and F. Pernuš. A heirarchical approach to elastic registration based on mutual information. *Image and Vision Computing*, 19:33–44, 2001.

[18] Holden M, D. L. G. Hill, E. R. E. Denton, J. M. Jarosz, T. C. S. Cox, T. Rohlfing, J. Goodey, and D. J. Hawkes. Voxel similarity measures for 3-D serial MR brain image registration. *IEEE Transactions on Medical Imaging*, 19(2):94–102, 2000.

[19] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Seutens. Multimodality image registration by maximisation of mutual information. *IEEE Transactions on Medical Imaging*, 16:187–197, 1997.

[20] D. C. Noll, F. E. Boada, and W. F. Eddy. A spectral approach to analyzing slice selection in planar imaging: Optimization for through-plane interpolation. *Magnetic Resonance in Medicine*, 38:151–160, 1997.

[21] C. A. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C. T. Chen. Accurate three-dimensional registration of CT, PET and MR images of the brain. *Journal of Computer Assisted Tomography*, 13:20–26, 1988.

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C (Second Edition).* Cambridge, Cambridge, 1992.

[23] C. Studholme, R. T. Constable, and J. S. Duncan. Accurate alignment of functional EPI data to anatomical MRI using a physics-based distortion model. *IEEE Transactions on Medical Imaging*, 19(11):1115–1127, 2000.

[24] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Multiresolution voxel similarity measures for MR-PET coregistration. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Proc. Information Processing in Medical Imaging*, pages 287–298, Dordrecht, The Netherlands, 1995. Kluwer Academic Publishers.

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

[26] J. Talairach and P. Tournoux. *Coplanar stereotaxic atlas of the human brain.* Thieme Medical, New York, 1988.

[27] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.

[28] P. Thévenaz and M. Unser. Optimization of mutual information for multiresolution image registration. *IEEE Transactions on Image Processing*, 9(12):2083–2099, 2000.

[29] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I – theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, 1993.

[30] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part II – efficient design and applications. *IEEE Transactions on Signal Processing*, 41(2):834–848, 1993.

[31] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximisation of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996.

[32] R. P. Woods. *Brain Warping*, chapter 20, pages 365–376. Academic Press, 1999.

[33] R. P. Woods. *Handbook of Medical Imaging*, chapter 33, pages 529–536. Academic Press, 2000.

[34] 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. *Journal of Computer Assisted Tomography*, 22(1):139–152, 1998.

[35] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17:536–546, 1993.

[36] T. D. Zuk and M. S. Atkins. A comparison of manual and automatic methods for registering scans of the head. *IEEE Transactions on Medical Imaging*, 15(5):732–744, 1996.