Chapter 2

Rigid Body Registration

Contents

2.1 Introduction

Rigid body registration is normally used for registering images of the same subject. This chapter describes methods of within subject registration for images of the same or different modalities.

For every image registration, the spatial transformation should be described by a set of parameters. In three dimensions, rigid registration requires 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 parameters describing a transformation are estimated. Then there is the *transformation*, where one of the images is transformed according to the set of parameters.

At its simplest, image registration involves estimating a mapping between a pair of images. One image is assumed to remain stationary (the target or template image), whereas the other (the source image) is spatially transformed to match it. In order to transform the source to match the target, it is necessary to determine a mapping from each voxel position ($\mathbf{x}$) in the target to a corresponding position ($\mathbf{y}$) in the source. The source is then resampled at the new positions. The vector function $\mathbf{y}$ can be thought of as a function of $\mathbf{x}$, and a set of transformation parameters $\mathbf{q}$ that are estimated in order to register the images.

This chapter will touch first on how rigid transformations are parameterised in terms of affine transformations. The next section explains how the images are transformed via the process of resampling, before the optimisation section explains how the best values for the parameters ($\mathbf{q}$) are estimated. The simplest form of within subject registration involves registering together two images of the same modality. A method for doing this is briefly described, before the final section describes a more complex technique for performing between modality registration.

2.2 Affine Transformations

Rigid body transformations are a subset of the more general affine 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:

$$\begin{aligned} y_1 &= m_{11}x_1 + m_{12}x_2 + m_{13}x_3 + m_{14} \\ y_2 &= m_{21}x_1 + m_{22}x_2 + m_{23}x_3 + m_{24} \\ y_3 &= m_{31}x_1 + m_{32}x_2 + m_{33}x_3 + m_{34} \end{aligned} \tag{2.1}$$

This mapping is often expressed as a simple matrix multiplication ($\mathbf{y} = \mathbf{M}\mathbf{x}$):

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \tag{2.2}$$

The elegance of formulating these transformations in terms of matrices is that several transformations can be combined simply by multiplying the matrices together to form a single matrix. This means that repeated resampling of data can be avoided when reorienting an image.

Translations

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

$$\mathbf{y} = \mathbf{x} + \mathbf{q} \tag{2.3}$$

In matrix terms, this transformation can be considered as:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \tag{2.4}$$

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:

$$\begin{array}{rcrcl} y_1 & = & cos(\theta)x_1 & + & sin(\theta)x_2 \\ y_2 & = & -sin(\theta)x_1 & + & cos(\theta)x_2 \end{array} \tag{2.5}$$

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. For simplicity, the 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:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(q_1) & sin(q_1) & 0 \\ 0 & -sin(q_1) & cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \tag{2.6}$$

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 in which the operations are performed is important. For example, a rotation about the first axis of $\pi/2$ radians followed by an equivalent rotation about the second axis would produce a very different result to that obtained if the order of the operations was reversed.

Zooms

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

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \tag{2.7}$$

A single zoom by a factor of -1 will flip the object (see Section 2.2.3). Two flips in different directions will merely rotate the object by π radians (a rigid body transformation). In fact, any affine transformation that has a negative determinant will render the object flipped.

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 rigid rotations. For a simple two dimensional case, a matrix encoding a rotation of θ radians about the origin (see Section 2.5) can be constructed by multiplying together three matrices that effect shears:

$$\begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & tan(\theta/2) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ sin(\theta) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & tan(\theta/2) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

This approach has been useful for rigid registration of MR images (Eddy *et al.*, 1996), and subsequently improved by a more efficient reformulation for three dimensional transformations (Cox & Jesmanowicz, 1999).

2.2.1 Parameterising a Rigid Body Transformation

When registering a pair of images together via a rigid body transformation, it is necessary to estimate six parameters that describe the rigid-body transformation matrix. There are many ways of parameterising a rigid body transformation in terms of six parameters ($\mathbf{q}$), but the parameterisation chosen here is:

$$\mathbf{M} = \mathbf{TR} \tag{2.9}$$

where:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

and:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(q_4) & sin(q_4) & 0 \\ 0 & -sin(q_4) & cos(q_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(q_5) & 0 & sin(q_5) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(q_5) & 0 & cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(q_6) & sin(q_6) & 0 & 0 \\ -sin(q_6) & cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

Extracting the 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:

$$\mathbf{R} = \begin{bmatrix} c_5c_6 & c_5s_6 & s_5 & 0 \\ -s_4s_5c_6 - c_4s_6 & -s_4s_5s_6 + c_4c_6 & s_4c_5 & 0 \\ -c_4s_5c_6 + s_4s_6 & -c_4s_5s_6 - s_4c_6 & c_4c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

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, providing that c_5 is not zero, then:

$$\begin{aligned} q_5 = & \quad sin^{-1}(r_{13}) \\ q_4 = & \quad atan2(r_{23}/cos(q_5), r_{33}/cos(q_5)) \\ q_6 = & \quad atan2(r_{12}/cos(q_5), r_{11}/cos(q_5)) \end{aligned} \tag{2.13}$$

where $atan2$ is the four quadrant inverse tangent. See Section 6.4 for more on decomposing affine transformations containing zooms and shears.

2.2.2 Working with Volumes of Differing or Anisotropic Voxel Sizes

Image voxel sizes need be considered when performing rigid body 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 millimetres. Rather than interpolating the images such that the voxels are cubic and have the same dimensions 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:

$$\mathbf{M_f} = \begin{bmatrix} 2.1 & 0 & 0 & -134.4 \\ 0 & 2.1 & 0 & -134.4 \\ 0 & 0 & 2.45 & -52.675 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

This transformation matrix maps voxel co-ordinates to a Euclidean space who's axes are parallel to those of the image and distances are measured in millimetres, with the origin at the centre of the image. 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 any co-registration is to determine the rigid body 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 $\mathbf{M_r M_f}$. 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 co-registered to image $\mathbf{g}$ in the same manner, then not only is there a mapping from $\mathbf{h}$ and $\mathbf{g}$ (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 $\mathbf{M_f}^{-1}\mathbf{M_g}\mathbf{M_g}^{-1}\mathbf{M_h}$).

2.2.3 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 2.1), where one system is a mirror image of the other. For example, the system used by the Talairach Atlas (Talairach & Tournoux, 1988) is right-handed, because the first dimension

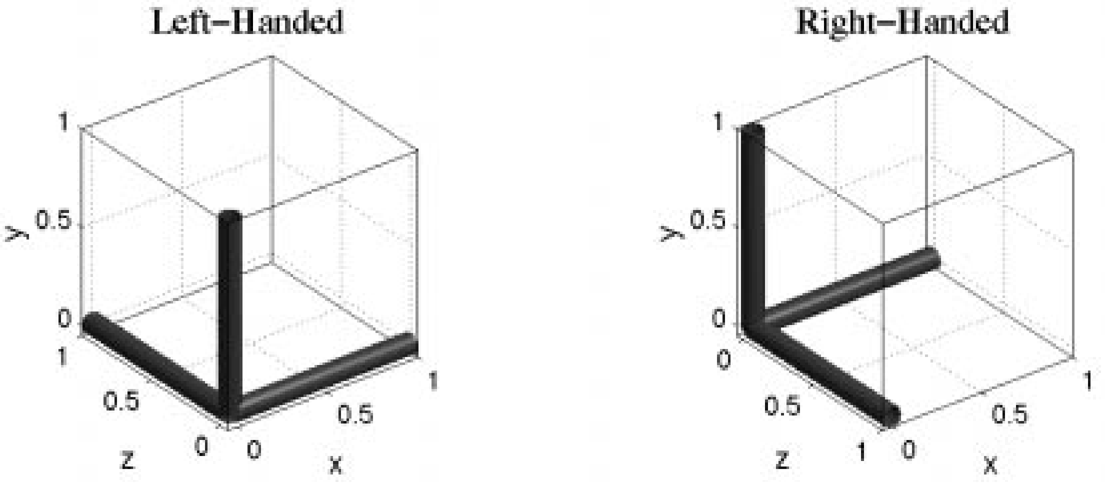

Figure 2.1: Left- and right-handed co-ordinate systems.

(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 that maps 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 reorienting brain image volumes. Consistency of the co-ordinate systems can be achieved by performing any reorientations using affine transformations, and checking the determinants of the transformation matrices.

2.3 Resampling Images

Once there is a mapping between the original and transformed co-ordinates of an image, it is necessary to resample the image in order to apply the spatial transform. Spatially transforming images 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 centres of voxels, so some form of interpolation is needed.

The simplest approach is to take the value of the closest neighbouring voxel. This is referred to as *nearest neighbour* or *zero-order hold* resampling. This has the advantage that the original voxel intensities are preserved, but the resulting image is degraded quite considerably.

Another approach is to use *tri-linear interpolation* (*first-order hold*) to resample the data. This is slightly slower than nearest neighbour, but the resulting images have a less “blocky” appearance. However, tri-linear interpolation has the effect of losing some high frequency information from the image.

Figure 2.2 will now be used to illustrate bilinear interpolation in two dimensions. Assuming

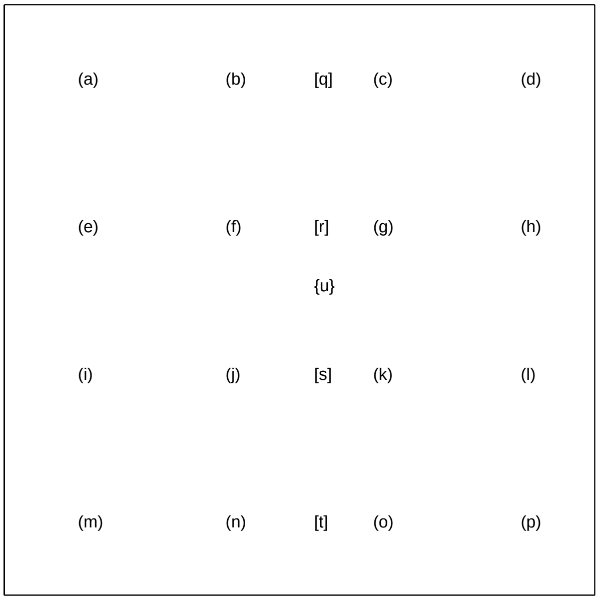

Figure 2.2: Illustration of image interpolation in two dimensions. Points a through to p represent the original regular grid of pixels. Point u is the point who's value is to be determined. Points q to t are used as intermediates in the computation.

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 resample is at u. The value at points r and s are first determined (using linear interpolation) as follows:

$$\begin{aligned} v_r &= \frac{(x_g - x_r)v_f + (x_r - x_f)v_g}{x_g - x_f} \\ v_s &= \frac{(x_k - x_s)v_j + (x_s - x_j)v_k}{x_k - x_j} \end{aligned} \tag{2.15}$$

Then v_u is determined by interpolating between v_r and v_s:

$$v_u = \frac{(y_u - y_s)v_r + (y_r - y_u)v_s}{y_r - y_s} \tag{2.16}$$

The extension of the approach to three dimensions is trivial.

Rather than using only the 8 nearest neighbours (in 3D) to estimate the value at a point, more neighbours can be used in order to fit a smooth function through the points, 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:

$$\mathbf{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & (x_b - x_a) & (x_b - x_a)^2 & (x_b - x_a)^3 \\ 1 & (x_c - x_a) & (x_c - x_a)^2 & (x_c - x_a)^3 \\ 1 & (x_d - x_a) & (x_d - x_a)^2 & (x_d - x_a)^3 \end{bmatrix}^{-1} \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \end{bmatrix} \tag{2.17}$$

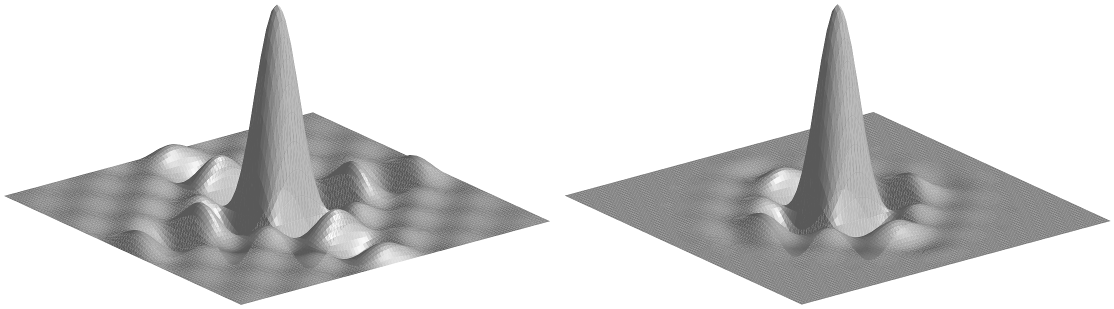

Figure 2.3: Sinc function in two dimensions, both with (right) and without (left) a Hanning window.

Then v_q can be determined from these coefficients by:

$$v_q = \begin{bmatrix} 1 & (x_q - x_a) & (x_q - x_a)^2 & (x_q - x_a)^3 \end{bmatrix} \mathbf{q} \tag{2.18}$$

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 Press *et al.*(1992) or Jain (1989)).

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 centred on the point to be resampled. 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 neighbours is used. Because the sinc function extends to infinity, it is often truncated by modulating with a Hanning window (see Figure 2.3). 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 neighbours would be:

$$\sum_{i=1}^{I} v_i \frac{\frac{sin(\pi d_i)}{\pi d_i} \frac{1}{2} \left(1 + cos\left(2\pi d_i / I\right)\right)}{\sum_{j=1}^{I} \frac{sin(\pi d_j)}{\pi d_j} \frac{1}{2} \left(1 + cos\left(2\pi d_j / I\right)\right)} \tag{2.19}$$

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

Sinc interpolation is slow when many neighbouring voxels are used. A slightly better alternative may be to use a Fourier interpolation method for effecting a rigid body transformation. In one dimension, a translation is simply a convolution with a translated delta function. For translations that are not whole numbers of pixels, the delta function is replaced by a sinc function centred 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 the rotations by a series of shears as described previously (Section 2.2). A shear simply involves translating different rows or columns of an image by different amounts, so each shear can be performed by a series of one dimensional convolutions in Fourier space. Alternatively, the method of rotating and translating using shears can also be done using a windowed sinc or polynomial interpolation. Each interpolation is in just one dimension, requiring much less computation than it would in three dimensions.

In addition to resampling images, many image registration methods also require the image gradients to be computed. This procedure is similar to the straightforward interpolation methods described above.

2.4 Optimisation

The objective of optimisation is to determine the values for a set of parameters for which some function of the parameters is minimised (or maximised). One of the simplest cases involves determining the optimum parameters for a model in order to minimise 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 optimisation is terminated when some convergence criterion is achieved (usually when χ^2 stops decreasing).

The image registration approach described here is essentially an optimisation. One image (the source image) is spatially transformed so that it matches another (the target image), by minimising χ^2. The parameters that are optimised are those that describe the spatial transformation (although there are often other nuisance parameters required by the model, such as intensity scaling parameters). For rigid registration, the algorithm chosen (Friston *et al.*, 1995c) is *Gauss-Newton* optimisation, and it is illustrated here:

Suppose that $b_i(\mathbf{q})$ is the function describing the difference between the source and target 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 $\mathbf{t}$:

$$b_i(\mathbf{q}-\mathbf{t}) \simeq b_i(\mathbf{q}) - t_1\frac{\partial b_i(\mathbf{q})}{\partial q_1} - t_2\frac{\partial b_i(\mathbf{q})}{\partial q_2}\dots \tag{2.20}$$

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 minimise $\sum_i b_i(\mathbf{q}-\mathbf{t})^2$:

$$\begin{bmatrix} \frac{\partial b_1(\mathbf{q})}{\partial q_1} & \frac{\partial b_1(\mathbf{q})}{\partial q_2} & \dots \\ \frac{\partial b_2(\mathbf{q})}{\partial q_1} & \frac{\partial b_2(\mathbf{q})}{\partial q_2} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \end{bmatrix} \simeq \begin{bmatrix} b_1(\mathbf{q}) \\ b_2(\mathbf{q}) \\ \vdots \end{bmatrix} \tag{2.21}$$

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

$$\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} - \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{2.22}$$

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. 2.22 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.

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 Press *et al.*(1992), Section 15.5). Another way of thinking about the optimisation 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 2.4).

2.5 Within Modality Image Registration

Within modality image registration has a number of uses, both within morphometry and for processing functional images. Morphometric studies sometimes involve looking at changes in brain shape over time, often to study the progression of a disease such as Altzheimers, or to monitor tumour 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 (Freeborough & Fox, 1998).

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 minimising 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 nonstationary variance in the images, a variance image could be computed

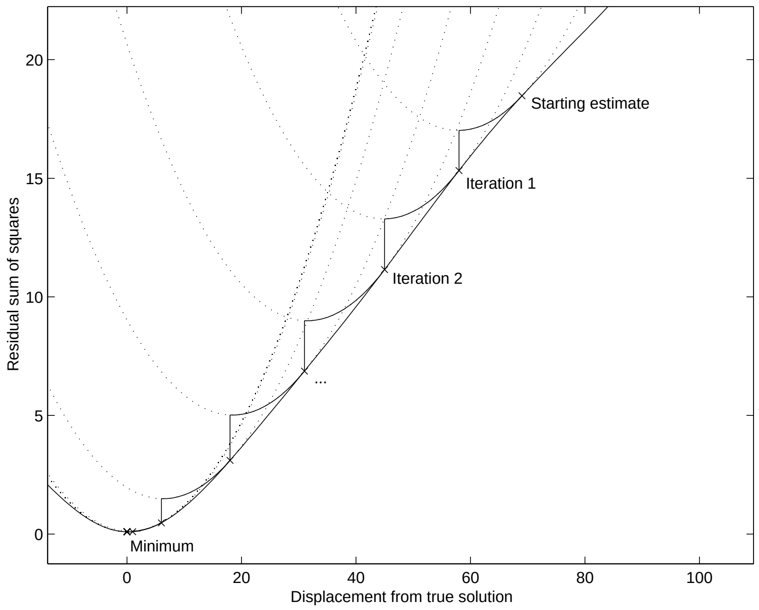

Figure 2.4: The optimisation can be thought of as fitting a series of quadratics to the error surface. Each parameter update is such that it falls at the minimum of the quadratic.

at the same time as the mean, in order to provide better weighting for the registration. Voxels with a lot of variance should be given lower weighting, whereas those with less variance should be weighted more highly.

2.5.1 Methods

To register a source image $\mathbf{f}$ to a reference image $\mathbf{g}$, a six parameter rigid body transformation (parameterised 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 $\mathbf{M}\mathbf{x}_i$, 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 optimised by minimising the sum of squared differences[1] between the images according to the algorithm described in Sections 2.2.1 and 2.4 (Eqn. 2.22). The function that is minimised is:

$$\sum_i \left(f(\mathbf{M}\mathbf{x}_i) - q_7 g(\mathbf{x}_i)\right)^2 \tag{2.23}$$

[1] Strictly speaking, it is the mean squared difference that is minimised, rather than the sum of squared differences. Inevitably, some values of $\mathbf{M}\mathbf{x}_i$ 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 $\mathbf{M}\mathbf{x}_i$ lie within the field of view of the images.

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 2.2.2). Vector $\mathbf{b}$ is generated for each iteration as:

$$\mathbf{b} = \begin{bmatrix} f(\mathbf{M}\mathbf{x}_1) - q_7 g(\mathbf{x}_1) \\ f(\mathbf{M}\mathbf{x}_2) - q_7 g(\mathbf{x}_2) \\ \vdots \end{bmatrix} \tag{2.24}$$

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

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f(\mathbf{M}\mathbf{x}_1)}{\partial q_1} & \frac{\partial f(\mathbf{M}\mathbf{x}_1)}{\partial q_2} & \cdots & \frac{\partial f(\mathbf{M}\mathbf{x}_1)}{\partial q_6} & -g(\mathbf{x}_1) \\ \frac{\partial f(\mathbf{M}\mathbf{x}_2)}{\partial q_1} & \frac{\partial f(\mathbf{M}\mathbf{x}_2)}{\partial q_2} & \cdots & \frac{\partial f(\mathbf{M}\mathbf{x}_2)}{\partial q_6} & -g(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{bmatrix} \tag{2.25}$$

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 minimising:

$$\sum_j \left(g(\mathbf{M}^{-1}\mathbf{y}_j) - q_7^{-1} f(\mathbf{y}_j)\right)^2 \tag{2.26}$$

In theory, a more robust solution could be achieved by simultaneously including the inverse transformation to make the registration problem symmetric (Woods *et al.*, 1998a). The cost function would then be:

$$\lambda_1 \sum_i \left(f(\mathbf{M}\mathbf{x}_i) - q_7 g(\mathbf{x}_i)\right)^2 + \lambda_2 \sum_j \left(g(\mathbf{M}^{-1}\mathbf{y}_j) - q_7^{-1} f(\mathbf{y}_j)\right)^2 \tag{2.27}$$

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:

$$\mathbf{b} = \begin{bmatrix} \lambda_1^{\frac{1}{2}}\left(f(\mathbf{M}\mathbf{x}_1) - q_7 g(\mathbf{x}_1)\right) \\ \lambda_1^{\frac{1}{2}}\left(f(\mathbf{M}\mathbf{x}_2) - q_7 g(\mathbf{x}_2)\right) \\ \vdots \\ \lambda_2^{\frac{1}{2}}\left(g(\mathbf{M}^{-1}\mathbf{y}_1) - q_7^{-1} f(\mathbf{y}_1)\right) \\ \lambda_2^{\frac{1}{2}}\left(g(\mathbf{M}^{-1}\mathbf{y}_2) - q_7^{-1} f(\mathbf{y}_2)\right) \\ \vdots \end{bmatrix} \tag{2.28}$$

and

$$\mathbf{A} = \begin{bmatrix} \lambda_1^{\frac{1}{2}}\frac{\partial f(\mathbf{M}\mathbf{x}_1)}{\partial q_1} & \cdots & \lambda_1^{\frac{1}{2}}\frac{\partial f(\mathbf{M}\mathbf{x}_1)}{\partial q_6} & -\lambda_1^{\frac{1}{2}} g(\mathbf{x}_1) \\ \lambda_1^{\frac{1}{2}}\frac{\partial f(\mathbf{M}\mathbf{x}_2)}{\partial q_1} & \cdots & \lambda_1^{\frac{1}{2}}\frac{\partial f(\mathbf{M}\mathbf{x}_2)}{\partial q_6} & -\lambda_1^{\frac{1}{2}} g(\mathbf{x}_2) \\ \vdots & \ddots & \vdots & \vdots \\ \lambda_2^{\frac{1}{2}}\frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_1)}{\partial q_1} & \cdots & \lambda_2^{\frac{1}{2}}\frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_1)}{\partial q_6} & \lambda_2^{\frac{1}{2}} q_7^{-2} f(\mathbf{y}_1) \\ \lambda_2^{\frac{1}{2}}\frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_2)}{\partial q_1} & \cdots & \lambda_2^{\frac{1}{2}}\frac{\partial g(\mathbf{M}^{-1}\mathbf{y}_2)}{\partial q_6} & \lambda_2^{\frac{1}{2}} q_7^{-2} f(\mathbf{y}_2) \\ \vdots & \ddots & \vdots & \vdots \end{bmatrix} \tag{2.29}$$

Symmetric formulation of registration problems is a theme that will be returned to in Chapter 4.

2.5.2 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 (Andersson *et al.*, 1995), but they are beyond the scope of this thesis.

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

- Interpolation error from the resampling algorithm used to transform the images can be one of the main sources of motion related artifacts. When the image series is resampled, it is important to use a very accurate interpolation method such as sinc or Fourier interpolation.
- 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 resampled, because it leads to errors at the edge of the brain that can not 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 (Noll *et al.*, 1997).
- 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 can not be corrected by a rigid body transformation (Jezzard & Clare, 1999).
- 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 leads to another reason why the images may not strictly obey the rules of rigid body motion.
- After a slice is magnetised, 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.
- Ghost artifacts in the 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 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.

These problems can not 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 (Friston *et al.*, 1996). 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.

2.6 Between Modality Image Registration

Co-registration of brain images of the same subject acquired in different modalities has proved itself to be useful in many areas, both in research and clinically. 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 minimise the sum of squared differences between them. However, for co-registration between modalities there is nothing quite as obvious to minimise.

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 (Pelizzari *et al.*, 1988). This method involved extracting brain surfaces of the two images, and then matching the surfaces together. Another method that has been widely used for a number of years for registering PET to MR images is AIR (Woods *et al.*, 1992). This method divides the MR images into a number of partitions based on intensity. The registration is based on minimising the standard deviation 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 laborious manual editing in order to remove any non-brain tissue.

More recently, the idea of matching images by maximising the *mutual information* in their histograms is becoming more widespread (Collignon *et al.*, 1995). For this elegant approach, the 2D histogram is normalised so that the bins integrate to unity. This is considered as an I by J matrix ($\mathbf{H}$, see Figure 3.7 for examples based on smooth images), and the registration involves maximising the following objective function (where element i,j of $\mathbf{H}$ is denoted by h_{ij}):

$$\sum_{j=1}^{J}\sum_{i=1}^{I} h_{ij} log\left(\frac{h_{ij}}{(\sum_{k=1}^{J} h_{ik})(\sum_{l=1}^{I} h_{lj})}\right) \tag{2.30}$$

Maximising mutual information is a very general approach, which has been successfully applied to the registration of a wide variety of imaging modalities.

The work developed here concentrates on a different fully automatic method of registering magnetic resonance (MR) images with positron emission tomography (PET) images, and on registering MR images from different scanning sequences. The assumption that it makes is that brains consist of two tissue types (grey and white matter) that can clearly be identified from the images.

2.6.1 Methods

This image co-registration method relies on images other than those that are to be registered ($\mathbf{f}$ and $\mathbf{g}$). These are template images of the same modalities ($\mathbf{t_f}$ and $\mathbf{t_g}$ respectively), and prior probability images of grey matter (GM), white matter (WM) and cerebro-spinal fluid (CSF). The template images and probability images conform to the same anatomical space, and examples of these are shown in Figure 3.6 on page 55.

The between modality co-registration is a three step approach that essentially reduces the problem to a series of within-modality approaches:

1. *Determine the affine transformations that map between the images and templates* by minimising the sum of squared differences between $\mathbf{f}$ and $\mathbf{t_f}$, and $\mathbf{g}$ and $\mathbf{t_g}$. These transformations are constrained such that only parameters that describe rigid body component are allowed to differ between the two registrations.

2. *Segment or partition the images using the probability images* and a modified *mixture model* algorithm (described in Chapter 5). The mapping between the probability images to images $\mathbf{f}$ and $\mathbf{g}$ having been determined in step 1.

3. *Co-register the image partitions* generated by the previous step using the rigid body transformations computed from step 1 as a starting estimate an using a within modality approach.

Determining the mappings from images to templates

It is possible to obtain a reasonable match of images of most normal brains to a template image using just a twelve parameter affine transformation. One can register image $\mathbf{g}$ to template $\mathbf{t_g}$, and similarly register $\mathbf{f}$ to $\mathbf{t_f}$ using this approach. These transformation matrices will be called $\mathbf{M_{gt}}$ and $\mathbf{M_{ft}}$ respectively. Thus a mapping from voxels in $\mathbf{g}$ to those in $\mathbf{f}$ is $\mathbf{M_f}^{-1}\mathbf{M_{ft}}\mathbf{M_{gt}}^{-1}\mathbf{M_g}$ (see Section 2.2.2 for a description of matrices $\mathbf{M_g}$ and $\mathbf{M_f}$). However, this affine transformation between $\mathbf{f}$ and $\mathbf{g}$ has not been constrained to be rigid body. This simple approach is modified in order to incorporate this constraint, by decomposing matrix $\mathbf{M_{gt}}$ into matrices that perform a rigid body transformation ($\mathbf{M_{gr}}$), and one that performs the scaling and shearing ($\mathbf{M_{ta}}$). i.e., $\mathbf{M_{gt}} = \mathbf{M_{gr}}\mathbf{M_{ta}}$. Similarly $\mathbf{M_{ft}} = \mathbf{M_{fr}}\mathbf{M_{ta}}$, where $\mathbf{M_{ta}}$ is common to both $\mathbf{M_{gt}}$ and $\mathbf{M_{ft}}$ so that the same zooms and shears are used for registering both images to their respective templates. Now the mapping becomes $\mathbf{M_f}^{-1}\mathbf{M_{fr}}(\mathbf{M_{ta}}\mathbf{M_{ta}}^{-1})\mathbf{M_{gr}}^{-1}\mathbf{M_g}$, and is a rigid body transformation. These matrices are parameterised by 18 elements of a vector $\mathbf{q}$.

$$
\begin{aligned}
\mathbf{M_{gr}} = {} & \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(q_4) & sin(q_4) & 0 \\ 0 & -sin(q_4) & cos(q_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\
& \begin{bmatrix} cos(q_5) & 0 & sin(q_5) & 0 \\ 0 & 0 & & 0 \\ -sin(q_5) & 0 & cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(q_6) & sin(q_6) & 0 & 0 \\ -sin(q_6) & cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{M_{fr}} = {} & \begin{bmatrix} 1 & 0 & 0 & q_7 \\ 0 & 1 & 0 & q_8 \\ 0 & 0 & 1 & q_9 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(q_{10}) & sin(q_{10}) & 0 \\ 0 & -sin(q_{10}) & cos(q_{10}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\
& \begin{bmatrix} cos(q_{11}) & 0 & sin(q_{11}) & 0 \\ 0 & 0 & & 0 \\ -sin(q_{11}) & 0 & cos(q_{11}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(q_{12}) & sin(q_{12}) & 0 & 0 \\ -sin(q_{12}) & cos(q_{12}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{M_{ta}} = {} & \begin{bmatrix} q_{13} & 0 & 0 & 0 \\ 0 & q_{14} & 0 & 0 \\ 0 & 0 & q_{15} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & q_{16} & q_{17} & 0 \\ 0 & 1 & q_{18} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2.31}
$$

The parameter set $\mathbf{q}$ can now be optimised in order to determine the transformations that minimise the sum of squares difference between the images and templates. The iterative optimisation method described in Section 2.4 is used, which generally converges within a few iterations. The chance of finding a local minimum is reduced by using smoothed data (typically by convolving with an 8mm full width at half maximum [FWHM] Gaussian kernel). Each iteration involves generating a linear approximation to the problem using Taylor's Theorem. This can be expressed as computing $\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} - \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$. The vector $\mathbf{q}^{(n)}$ contains the parameter estimates at iteration n, and vector $\mathbf{b}$ contains the differences between the templates and the images that have been spatially transformed according to the latest parameter estimates. $\mathbf{A}$ is a matrix of derivatives of $\mathbf{b}$, with respect to changes to each element of $\mathbf{q}$. For the purpose of this optimisation, two matrices, $\mathbf{M_1} = \mathbf{M_f}^{-1}\mathbf{M_{ft}}^{-1}\mathbf{M_t}$ and $\mathbf{M_2} = \mathbf{M_g}^{-1}\mathbf{M_{gt}}^{-1}\mathbf{M_t}$, are defined:

$$
\mathbf{b} = \begin{bmatrix} \lambda_1^{\frac{1}{2}}\left(f(\mathbf{M_1}\mathbf{x}_1) - q_{19}t_f(\mathbf{x}_1)\right) \\ \lambda_1^{\frac{1}{2}}\left(f(\mathbf{M_1}\mathbf{x}_2) - q_{19}t_f(\mathbf{x}_2)\right) \\ \vdots \\ \lambda_2^{\frac{1}{2}}\left(g(\mathbf{M_2}\mathbf{x}_1) - q_{20}t_g(\mathbf{x}_1)\right) \\ \lambda_2^{\frac{1}{2}}\left(g(\mathbf{M_2}\mathbf{x}_2) - q_{20}t_g(\mathbf{x}_2)\right) \\ \vdots \end{bmatrix}
\tag{2.32}
$$

$$\mathbf{A} = \begin{bmatrix} \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{M_1 x}_1)}{\partial q_1} & \dots & 0 & \dots & \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{M_1 x}_1)}{\partial q_{13}} & \dots & -\lambda_1^{\frac{1}{2}} t_f(\mathbf{x}_1) & 0 \\ \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{M_1 x}_2)}{\partial q_1} & \dots & 0 & \dots & \lambda_1^{\frac{1}{2}} \frac{\partial f(\mathbf{M_1 x}_2)}{\partial q_{13}} & \dots & -\lambda_1^{\frac{1}{2}} t_f(\mathbf{x}_2) & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \; \vdots & \vdots & \\ 0 & \dots & \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M_2 x}_1)}{\partial q_7} & \dots & \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M_2 x}_1)}{\partial q_{13}} & \dots & 0 & -\lambda_2^{\frac{1}{2}} t_g(\mathbf{x}_1) \\ 0 & \dots & \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M_2 x}_2)}{\partial q_7} & \dots & \lambda_2^{\frac{1}{2}} \frac{\partial g(\mathbf{M_2 x}_2)}{\partial q_{13}} & \dots & 0 & -\lambda_2^{\frac{1}{2}} t_g(\mathbf{x}_2) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \; \vdots & \vdots & \end{bmatrix} \tag{2.33}$$

The parameters describing the non-rigid transformations (q_{13} to q_{18}) could in theory be derived from either $\mathbf{f}$ or $\mathbf{g}$. In practice, a better solution is obtained by estimating these parameters using both images, and by biasing the result so that the image that fits the template better has a greater influence over the parameter estimates. This is achieved by weighting the rows of $\mathbf{A}$ and $\mathbf{b}$ that correspond to the different images. The weights (λ_1 and λ_2) are derived from the residual variance between the template and source images, obtained from the previous solution for $\mathbf{q}$. These are:

$$\lambda_1 = \frac{\nu_1}{\sum_{i=1}^{I}(f(\mathbf{M_1 x}_i) - q_{19} t_f(\mathbf{x}_i))^2} \text{ and } \lambda_2 = \frac{\nu_2}{\sum_{i=1}^{I}(g(\mathbf{M_2 x}_i) - q_{20} t_g(\mathbf{x}_i))^2} \tag{2.34}$$

where ν_1 and ν_2 are the degrees of freedom for the two parts of the problem. Also note that this first step is made more robust by including the regularisation that will be described in the next chapter.

Once the optimisation has converged to the final solution, the rigid body transformation that approximately maps between $\mathbf{g}$ and $\mathbf{f}$ can be obtained, and also the affine transformation matrices that map between the source images and templates. These are used by the next step.

Partitioning the Images

The result of the previous step includes affine mappings between source and template images, which are used to assist image segmentation by allowing prior probability images GM, WM and CSF to be automatically overlaid on to the source images. The extraction of GM and WM proceeds as described in Chapter 5 (although no correction for image intensity nonuniformity is included for the evaluations). The result of the partitioning are images representing the probability of the voxels belonging to each tissue class. All voxel intensities are between zero and one, and most lie close to one or the other extreme.

Co-registering the Image Partitions

The previous step produces images of GM and WM from the original images $\mathbf{f}$ and $\mathbf{g}$. These image partitions can then be simultaneously co-registered together to produce the final solution.

This optimisation stage only needs to search for the six parameters that describe a rigid body transformation. A voxel-to-voxel affine transform matrix $\mathbf{M}$ is defined by $(\mathbf{M_f}^{-1}\mathbf{M_{fg}}^{-1}\mathbf{M}_\mathrm{g})$, where the rigid body transformation matrix $\mathbf{M_{fg}}$ is parameterised by q_1 to q_6. Starting estimates for $\mathbf{q}$ are obtained by extracting them from the rigid transformation matrix $\mathbf{M_{gr}}\mathbf{M_{fr}}^{-1}$ as described in Section 2.2. Convergence should be achieved within a few iterations because of the good starting estimates obtained from the first step. No scaling parameters are needed, because the probability images derived from $\mathbf{f}$ have similar intensities to those derived from $\mathbf{g}$. Again, the method described in Section 2.4 is used to optimise the parameters, where (using notation where $p_{g1}(\mathbf{x}_2)$ means 'probability of voxel at $\mathbf{x}_2$ from image $\mathbf{g}$ belonging to cluster 1') $\mathbf{b}$ and $\mathbf{A}$ are defined by:

$$\mathbf{b} = \begin{bmatrix} p_{f1}(\mathbf{Mx}_1) - p_{g1}(\mathbf{x}_1) \\ p_{f1}(\mathbf{Mx}_2) - p_{g1}(\mathbf{x}_2) \\ \vdots \\ p_{f2}(\mathbf{Mx}_1) - p_{g2}(\mathbf{x}_1) \\ p_{f2}(\mathbf{Mx}_2) - p_{g2}(\mathbf{x}_2) \\ \vdots \end{bmatrix} \tag{2.35}$$

$$\mathbf{A} = \begin{bmatrix} \frac{\partial p_{f1}(\mathbf{Mx}_1)}{\partial q_1} & \frac{\partial p_{f1}(\mathbf{Mx}_1)}{\partial q_2} & \cdots & \frac{\partial p_{f1}(\mathbf{Mx}_1)}{\partial q_6} \\ \frac{\partial p_{f1}(\mathbf{Mx}_2)}{\partial q_1} & \frac{\partial p_{f1}(\mathbf{Mx}_2)}{\partial q_2} & \cdots & \frac{\partial p_{f1}(\mathbf{Mx}_2)}{\partial q_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial p_{f2}(\mathbf{Mx}_1)}{\partial q_1} & \frac{\partial p_{f2}(\mathbf{Mx}_1)}{\partial q_2} & \cdots & \frac{\partial p_{f2}(\mathbf{Mx}_1)}{\partial q_6} \\ \frac{\partial p_{f2}(\mathbf{Mx}_2)}{\partial q_1} & \frac{\partial p_{f2}(\mathbf{Mx}_2)}{\partial q_2} & \cdots & \frac{\partial p_{f2}(\mathbf{Mx}_2)}{\partial q_6} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \tag{2.36}$$

The final solution is obtained after this co-registration. It is now possible to map voxel $\mathbf{x}$ of image $\mathbf{g}$, to the corresponding voxel $\mathbf{Mx}$ of image $\mathbf{f}$. Examples of PET/T1-MRI and T1/T2-MRI co-registration using this approach are illustrated in figures 2.5 and 2.6.

2.6.2 Evaluation

The co-registration methods were evaluated for PET to T1 weighted MRI, using data from the "Evaluation of Retrospective Image Registration" project (National Institutes of Health, Project Number 1 R01 NS33926-01, Principal Investigator, J. Michael Fitzpatrick, Vanderbilt University, Nashville, TN.) (West *et al.*, 1996; West *et al.*, 1997). This involved obtaining both PET and MRI data from Vanderbilt University, and performing inter-modality registrations on the volumes. Fiducial markers during the acquisition of these datasets enabled investigators at Vanderbilt to know the true registration parameters, but any visible traces of these markers had been removed from the images prior to their distribution to other investigators.

Registrations were performed on 11 volume images. Four of the images had been geometry distortion corrected at Vanderbilt using their own software that uses a pair of distorted images acquired with reversed readout gradients. Seven of the images had not been corrected. The registrations were done using only the first step of the registration process (constrained simultaneous affine registration) and also using all the steps. The evaluations were performed on a Sun SPARC Ultra 2, using an implementation of the method written in C and Matlab (from The Mathworks, Natick, Mass., USA). The starting estimates for the registration parameters matched the centres of each volume together, and assumed that the images were in the same orientation. No manual

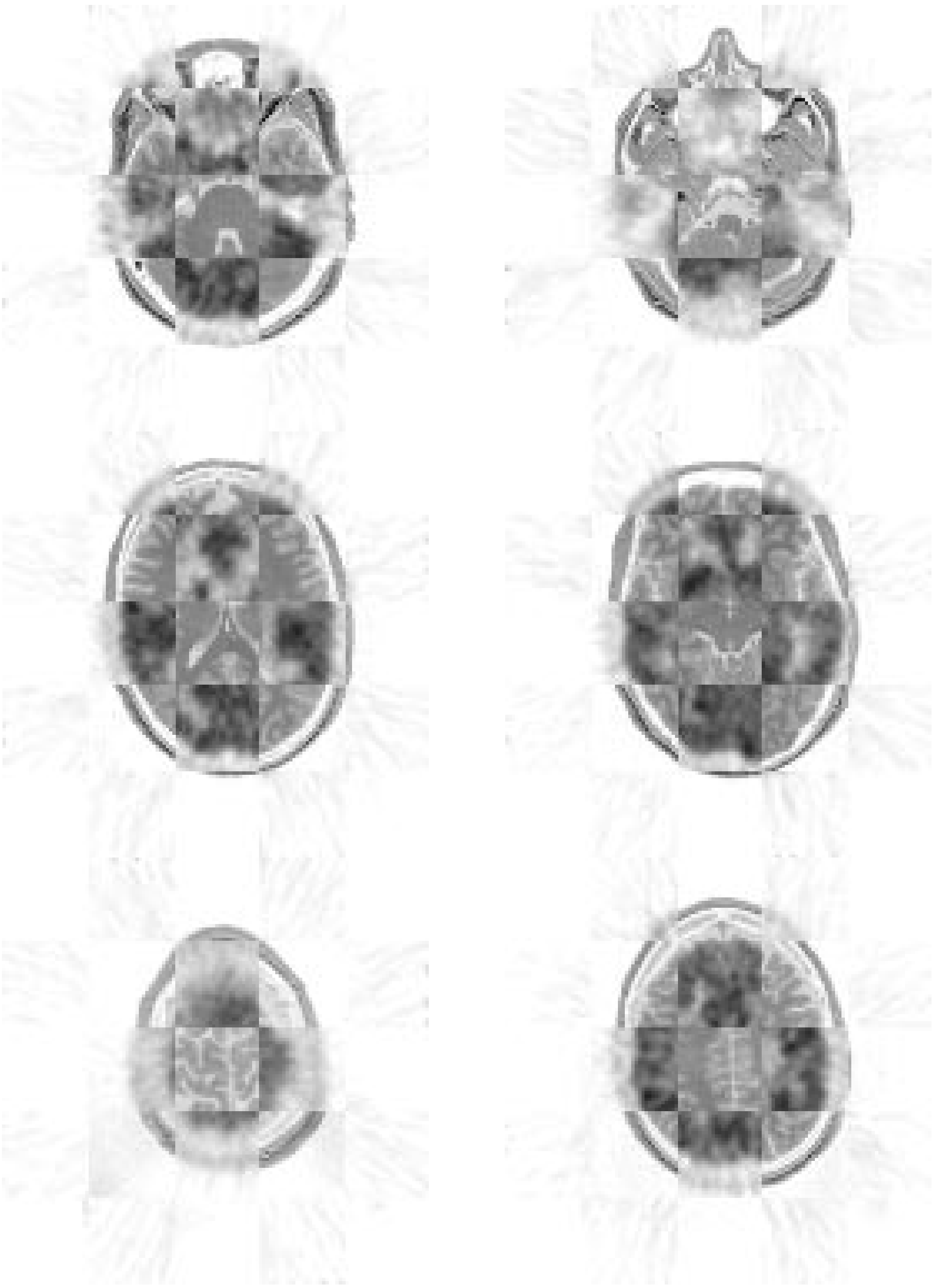

Figure 2.5: An example of PET-MRI co-registration, achieved using the techniques described here.

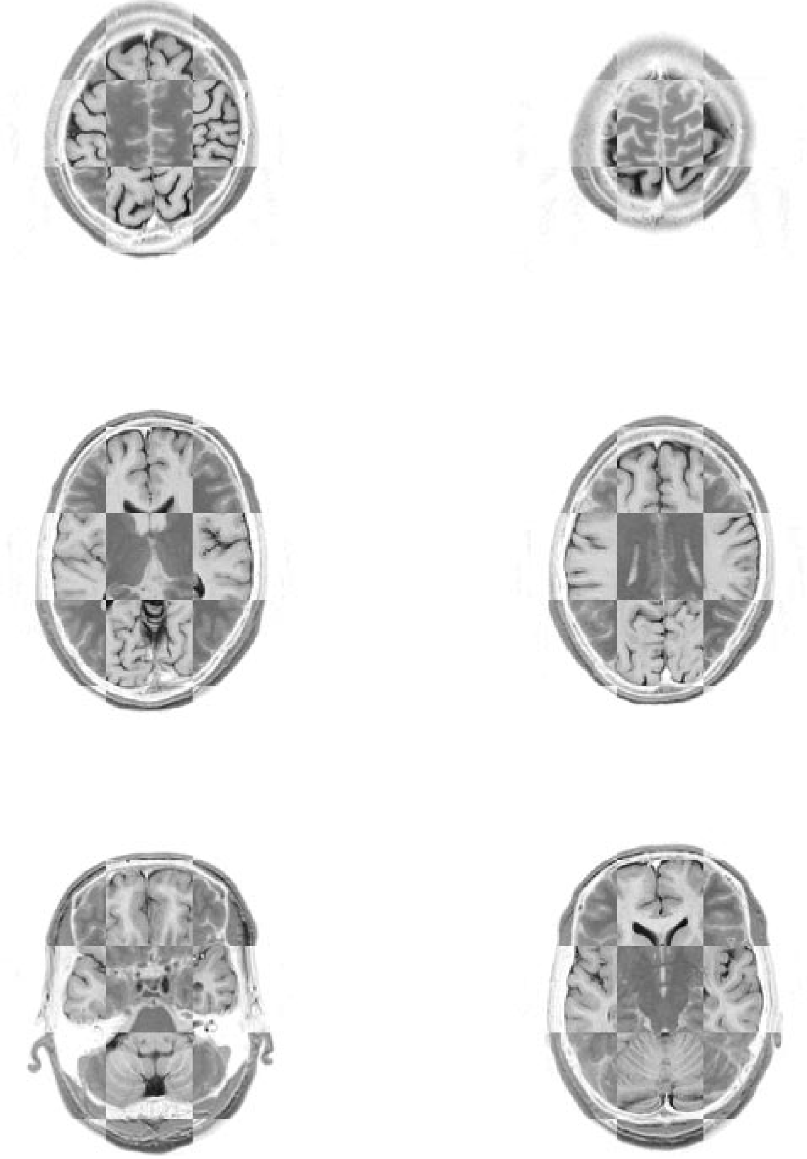

Figure 2.6: An example of T1-weighted and T2-weighted MR images of the same subject registered using the techniques described here.

	First step only		All three steps	
	Uncorrected	Corrected	Uncorrected	Corrected
mean error (mm)	5.57	3.77	4.14	3.20
median error (mm)	5.11	3.17	4.20	3.36
maximum error (mm)	11.62	8.54	7.46	5.76
N	7	4	7	4

Table 2.1: Errors for PET-MRI registration. Errors are presented, for both the uncorrected and the distortion corrected MR images. The results in the left hand column were derived after using only the first step of the registration process. The right hand column shows the results of using all three steps of the registration process.

	BA	CO	HA	HI	MAI	MAL
median error uncorrected	4.6	3.6*	2.8*	3.2*	3.5*	4.2*
median error corrected	3.2*	2.8*	3.6	2.5*	3.9	3.6
maximum error uncorrected	11.5	12.7	12.1	9.3	10.6	8.5
maximum error corrected	6.0	3.7*	17.7	6.0	7.7	8.4

	NO	PE	RO3	RO4	WO1	WO2
median error uncorrected	3.6*	2.9*	4.0*	3.4*	2.3*	3.1*
median error corrected	3.9	2.8*	3.8	3.6	2.0*	2.0*
maximum error uncorrected	11.4	10.0	9.4	5.9*	5.8*	6.0*
maximum error corrected	14.2	7.9	7.3	8.9	4.3*	5.0*

Table 2.2: Errors for PET-MRI registration from other methods. Asterisked values indicate results from methods that performed better than or the same as the one presented here. Method NO involved matching manually specified landmark pairs. Methods BA, HA, MAI, MAL, PE, RO3 and RO4 involved matching surfaces, contours or edges. WO1 and WO2 both used the AIR software (Woods *et al.*, 1992), but with different amounts of manual editing. CO and HI were based on maximising mutual information.

intervention was involved. The first step of the registration process (including the initial smoothing) took an average of 66 seconds, whereas the complete three step registration required about 350 seconds.

The resulting parameter estimates were communicated back to Vanderbilt, where their accuracy was evaluated. The results presented in Table 2.1 are the mean, median and maximum errors for the registration, and can be compared directly with those in Table 2.2 (taken from West *et al.*(1997)). They show that the first step of the process quickly registered the images to within about 6 mm, before the remaining steps further refined the parameters. The median registration errors using the current approach tended to be slightly larger than those from most other methods, although the maximum errors were generally smaller. Registration using AIR was found to produce the most consistently accurate solutions, but this requires initial manual pre-processing of the images.

Although the accuracy of the registrations was found to be comparable with the other techniques evaluated by West *et al.*, the data used in the evaluation did violate a number of the assumptions made by the current approach. Image intensity nonuniformity and low grey/white matter contrast resulted in a considerable amount of tissue misclassification of the MR images. Also, the assumption that brain tissue can be broadly classified as grey or white matter was complicated by the presence of tumours, which were classified as grey matter in the MR images, and white matter in the PET images. This would be expected to introduce additional registration errors, because the final step is based upon matching corresponding image partitions together. The registration should be much more accurate for images where the assumptions hold, as has been shown by more recent evaluations of SPECT and MRI co-registration methods (Barnden *et al.*, 2000).

In summary, the current technique is valid in relation to existing techniques. Unlike some of the existing approaches, the present method does not require manual intervention. The only occasional intervention that may be needed is to provide starting estimates to the first step. The procedure has so far been successfully applied to registering T1 weighted MRI to PET (blood flow), T1 to T2 weighted MRI, and T2 weighted MRI to PET.