

# Spatial Transformation of Images

John Ashburner & Karl J. Friston

*The Wellcome Department of Cognitive Neurology, Institute of Neurology, Queen  
Square, London WC1N 3BG, UK*

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Overview . . . . .	3
<b>2</b>	<b>The Underlying Principles</b>	<b>4</b>
2.1	Resampling Images . . . . .	4
2.2	Smoothing . . . . .	7
2.3	Affine Transformations . . . . .	8
2.3.1	Rigid Body Transformations . . . . .	8
2.4	Optimisation . . . . .	9
2.5	Intensity Transformations . . . . .	10
<b>3</b>	<b>Within Modality Image Co-registration</b>	<b>11</b>
3.1	Simple Affine Registration . . . . .	12
3.2	Constraining to be Rigid Body . . . . .	13
<b>4</b>	<b>Between Modality Image Co-registration (and Partitioning)</b>	<b>14</b>
4.1	Determining the mappings from images to templates. . . . .	15
4.2	Partitioning the images. . . . .	16
4.3	Co-registering the image partitions. . . . .	20
4.3.1	An alternative implementation for low resolution images. . . . .	20
4.4	Discussion. . . . .	21
<b>5</b>	<b>Affine Spatial Normalisation</b>	<b>22</b>
5.1	A Bayesian Approach. . . . .	22
5.2	Estimating $\mathbf{C}$ . . . . .	24
5.3	Estimating $\mathbf{p}_0$ and $\mathbf{C}_0$ . . . . .	25
5.4	Incorporating the Bayesian Approach into the Optimisation. . . . .	26
5.4.1	Stopping Criterion. . . . .	26
<b>6</b>	<b>Non-linear Spatial Normalisation</b>	<b>26</b>
6.1	A Basis Function Approach . . . . .	27
6.2	A Fast Algorithm . . . . .	28
6.3	Regularisation - a Bayesian-like approach . . . . .	31
6.4	Discussion . . . . .	32
<b>7</b>	<b>Summary</b>	<b>34</b>
	<b>References</b>	<b>35</b>

# 1 Introduction

This chapter describes how to identify the spatial and intensity transformations that map one image onto another. A general technique is presented that implements non-linear spatial (stereotactic) normalisation and image realignment.

Spatial transformations are important in many aspects of functional image analysis. In functional imaging, particularly for functional magnetic resonance imaging (fMRI), the signal changes due to any haemodynamic response can be small compared to signal changes that can result from subject motion, so prior to performing the statistical tests, it is important that the images are as closely aligned as possible. Subject head movement in the scanner can not be completely eliminated, so motion correction needs to be performed as a preprocessing step on the image data. The first step in the correction is image registration, which involves determining the parameter values for a rigid body transformation that optimise some criteria for matching each image with a reference image (see Section 3). Following the registration, the images are transformed by resampling according to the determined parameters.

Sometimes it is desirable to warp images from a number of individuals into roughly the same standard space to allow signal averaging across subjects. Unlike the case for positron emission tomography (PET) images, it is not necessary to combine fMRI data from a number of subjects to achieve a significant activation signal. However, since different people may have different strategies for performing tasks in the scanner, spatial normalisation of the images is useful for determining what happens generically over individuals. A further advantage of using spatially normalised images is that activation sites can be reported according to their Euclidian coordinates within a standard space (Fox, 1995). The most commonly adopted coordinate system within the brain imaging community is that described by Talairach & Tournoux (1988). The normalisation usually begins by matching the brains to a template image using an affine transformation (see Section 5), followed by introducing non-linear deformations described by a number of smooth basis functions (Friston *et al.*, 1995b) (see Section 6). Matching is only possible on a coarse scale, since there is not necessarily a one-to-one mapping of the cortical structures between different brains. Because of this, the images are smoothed prior to the statistical analysis in a multi-subject study, so that corresponding sites of activation from the different brains are superimposed (see Section 2.2).

For studies of a single subject, the sites of activation can be accurately localised by superimposing them on a high resolution structural image of the subject (typically a  $T_1$  weighted MRI). This requires the registration of the functional images with the structural image. As in the case of movement correction, this is normally performed by optimising a set of parameters describing a rigid body transformation, but the matching criterion needs to be more complex since the images are often acquired using different modalities or MR contrasts (see Section 4). A further use for this registration is that a more precise spatial normalisation can be achieved by computing it from a more detailed structural image. With the functional and structural images in register, the computed warps can be applied to the functional images.

## 1.1 Overview

The next sections of this chapter are arranged as follows:

- *The Underlying Principles* gives a brief introduction of some of the concepts that are used throughout the rest of the chapter.
  - *Resampling Images* describes a number of different interpolation methods that can be used to spatially transform images.
  - *Smoothing* is included because image registration can be performed more easily on images that are smooth.
  - *Affine Transformations* are among the commonest of spatial transformations of images. Rigid body transformations (which are a subset of affine transformations) are necessary for co-registering images of the same subject, and affine transformations are also the first step in spatial normalisation.
  - *Optimisation* is necessary to determine the best parameters for matching images together. The optimisation framework that is used throughout the rest of the chapter is introduced in this subsection.
  - The main sections concentrate on the nature of the spatial transformations required to match images. However, occasionally, some form of *Intensity Transformations* of one of the images are required in order to achieve a better fit.
- *Within Modality Image Co-registration* is probably the simplest form of image registration. It involves finding the best six parameter rigid body transformation to minimise the difference between two images of the same subject. This section introduces a basic method for performing affine registrations, and shows how to constrain the registration to be rigid body.
- *Between Modality Image Co-registration* again involves rigid body transformations, but in this case - since the images to be matched appear completely different - different matching strategies need to be used. This section includes a description of an image classification method, whereby gray matter, white matter and CSF are automatically identified in images of different modalities. The classification is important to this section because the registration ultimately relies on matching homologous tissue classes together.
- *Affine Spatial Normalisation* describes the first step involved in registering images of different subjects into roughly the same co-ordinate system. Unlike the previous two sections - where the images to be matched together are from the same subject - zooms and shears are needed to register heads of different shapes and sizes. The main new idea introduced in this section is about how to incorporate prior knowledge of the variability of head sizes into the registration.
- *Non-linear Spatial Normalisation* is about correcting for gross differences in head shapes that can not be accounted for by affine normalisation alone. These non-linear warps are modelled by linear combinations of smooth basis functions, and a fast algorithm for determining the optimum combination of basis functions is described.

## 2 The Underlying Principles

Spatial transformations can be broadly divided into *label based* and *non-label based*. Label based techniques identify homologous spatial structures, features or landmarks in two images and find the transformations that best superpose the labeled points. These transformations can be linear (Pelizzari *et al.*, 1988) or non-linear (e.g., thin plate splines (Bookstein, 1989)). Non-label based approaches identify a spatial transformation that minimises some index of the difference between an object and a reference image, where both are treated as unlabeled continuous processes. Again these can be linear, e.g., principal axes (Alpert *et al.*, 1990); image realignment (Woods *et al.*, 1992; Collins *et al.*, 1994b; Lange, 1994) or non-linear e.g., plastic transformation (Friston *et al.*, 1991; Collins *et al.*, 1994a) with some interesting developments using neural nets (Kosugi *et al.*, 1993).

Without any constraints it is of course possible to transform any image such that it matches another exactly. The issue is therefore less about the nature of the transformation and more about defining constraints under which a transformation is effected. The validity of a transformation can usually be reduced to the validity of these constraints. The first tenet of the general approach described here is that the constraints are explicit, reasonable and operationally specified. The reliability of label-based approaches is limited by the reproducibility of labeling. The second key aspect of our approach is therefore that it is non label-based and automatic.

There are two steps involved in registering images together. There is the *registration* itself, whereby the parameters describing a transformation are determined. Then there is the *transformation*, where one of the images is transformed according to the set of parameters. This section will first touch on how the images are transformed via the process of resampling, before briefly describing how smoothing (low pass filtering) is performed in *SPM*. One of the more common image transformations (the affine transformation) is then described, and finally the general principal of how the parameters describing the transformations are automatically determined.

### 2.1 Resampling Images

Once there is a mapping between the original and transformed coordinates of an image, it is necessary to resample the image in order to apply the spatial transform. 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 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 can be 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.

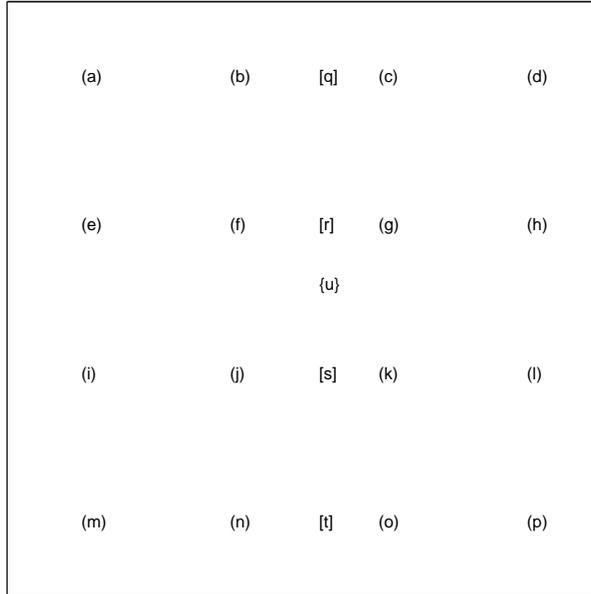


Figure 1: Illustration of image interpolation in two dimensions. Points  $a$  through to  $p$  represent the original regular grid of pixels. Point  $u$  is the point whose value is to be determined. Points  $q$  to  $t$  are used as intermediates in the computation.

Figure 1 will now be used to illustrate bi-linear interpolation in two dimensions. Assuming that there is a regular grid of pixels at coordinates  $x_a, y_a$  to  $x_p, y_p$ , having intensities  $v_a$  to  $v_p$ , and that the point to resample is at  $t$ . The value at points  $r$  and  $s$  are first determined (using linear interpolation) as follows:

$$v_r = \frac{(x_g - x_r)v_f + (x_r - x_f)v_g}{x_g - x_f}$$

$$v_s = \frac{(x_k - x_s)v_j + (x_s - x_j)v_k}{x_k - x_j}$$

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.

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 point. *Polynomial interpolation* is one such approach (zero-order and first-order hold interpolation are simply low order polynomial interpolations). We now illustrate 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:

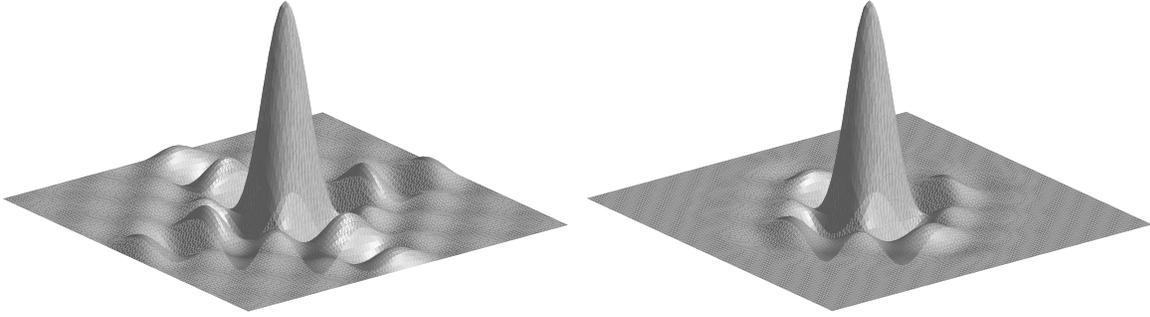


Figure 2: Sinc function in two dimensions, both with (right) and without (left) a Hanning window.

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

Then  $v_q$  can be determined from these coefficients by:

$$v_q = (1 \quad (x_q - x_a) \quad (x_q - x_a)^2 \quad (x_q - x_a)^3) \mathbf{q}$$

To determine  $v_u$ , a similar polynomial would be fitted through points  $q$ ,  $r$ ,  $s$  and  $t$ . Polynomial interpolation is normally performed using *Lagrange polynomials*. See (Press *et al.*, 1992) or (Jain, 1989) for more information on this or on interpolation in general.

The optimum method of transforming images without interpolation artifact is to do it in Fourier space (Eddy *et al.*, 1996). However, the rigid body transformations implemented in SPM are performed in real space. The interpolation method that gives results closest to Fourier interpolation is *sinc* interpolation. 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. Since the sinc function extends to infinity, it is truncated by modulating with a Hanning window (see Figure 2). 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:

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

where  $d_i$  is the distance from the center of the  $i$ th voxel to the point to be sampled, and  $v_i$  is the value of the  $i$ th voxel. This form of sinc interpolation is the preferred higher order interpolation method within *SPM*.

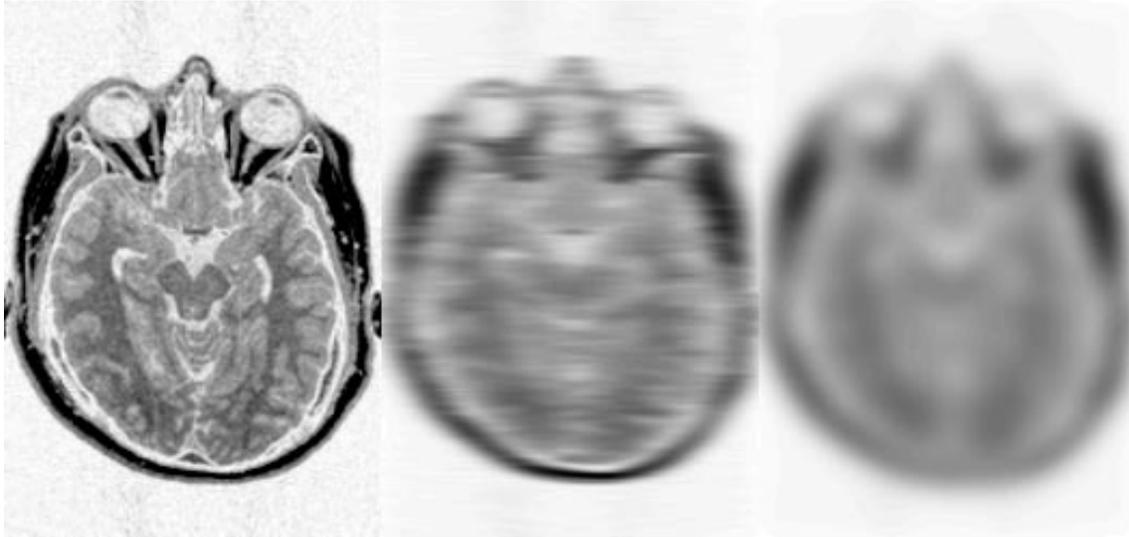


Figure 3: Convolution with a two dimensional Gaussian. The original image (left) is convolved horizontally (center), and then this image is convolved vertically (right).

## 2.2 Smoothing

Image registration is normally performed on smoothed images (for reasons that will be mentioned in Section 2.4). It is also important to smooth prior to the statistical analysis of a multi-subject experiment. Because the spatial normalisation can never be exact, homologous regions in the brains of the different subjects can not be precisely registered. The smoothing has the effect of ‘spreading out’ the different areas, and reducing the discrepancy.

The smoothing used is a discrete convolution with a Gaussian kernel. The amplitude of a Gaussian at  $j$  units away from the center is defined by:

$$g_j = \frac{e^{-\frac{j^2}{2s^2}}}{\sqrt{2\pi s^2}}$$

where the parameter  $s$  is defined by  $\frac{FWHM}{\sqrt{8\ln(2)}}$ , where  $FWHM$  is the *full width at half maximum* of the Gaussian. The convolution of function  $\mathbf{s}$  with  $\mathbf{g}$  to give the convolved function  $\mathbf{t}$  is performed as:

$$t_i = \sum_{j=-d}^d s_{(i-j)} g_j$$

In *SPM*, the value of  $d$  in the above expression will represent a kernel length of about six FWHMs. Beyond this distance, the magnitude of the function can be considered almost negligible.

To convolve an image with a two dimensional Gaussian, the image is first convolved in one direction, and then the result is convolved in the other (see Figure 3). A three dimensional convolution is the same, except for an additional convolution in the third dimension.

## 2.3 Affine Transformations

One of the simplest and well defined of spatial transformations is the affine transformation. For each point  $(x_1, x_2, x_3)$  in an image, a mapping can be defined into the coordinates of another space  $(y_1, y_2, y_3)$ . This is simply expressed as:

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

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

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

The elegance of formulating these transformations in terms of matrices is that several transformations can be combined by simply multiplying the matrices together to form a single matrix.

### 2.3.1 Rigid Body Transformations

Rigid body transformations, necessary to co-register images of the same subject together, are a subset of the more general affine transformations. In three dimensions a rigid body transformation can be defined by six parameters. These parameters are, typically, three translations (shifts) and three rotations about orthogonal axes. A matrix that implements the translation is:

$$\begin{pmatrix} 1 & 0 & 0 & x_{trans} \\ 0 & 1 & 0 & y_{trans} \\ 0 & 0 & 1 & z_{trans} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrixes that carry out rotations ( $\Theta$ ,  $\Phi$  and  $\Omega$  - in radians) about the  $X$ ,  $Y$  and  $Z$  axes respectively are:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta) & \sin(\Theta) & 0 \\ 0 & -\sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} \cos(\Phi) & 0 & \sin(\Phi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\Phi) & 0 & \cos(\Phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} \cos(\Omega) & \sin(\Omega) & 0 & 0 \\ -\sin(\Omega) & \cos(\Omega) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

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

Voxel sizes of images need to be considered in order to register them with a rigid body transformation. 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 Euclidian space is used, where measures of distances are expressed in millimeters. 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 coordinates into this Euclidian space. For example, if image  $\mathbf{f}$  is of size  $128 \times 128 \times 43$  and has voxels that are  $2.1mm \times 2.1mm \times 2.45mm$ , we can define the following matrix:

$$\mathbf{M}_{\mathbf{f}} = \begin{pmatrix} 2.1 & 0 & 0 & -134.4 \\ 0 & 2.1 & 0 & -134.4 \\ 0 & 0 & 2.45 & -52.675 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This transformation matrix maps voxel coordinates to a Euclidian 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. A similar matrix can be defined for  $\mathbf{g}$  ( $\mathbf{M}_{\mathbf{g}}$ ).

The objective of any co-registration is to determine the rigid body transformation that maps the coordinates of image  $\mathbf{f}$ , to that of  $\mathbf{g}$ . To accomplish this, a rigid body transformation matrix  $\mathbf{M}_{\mathbf{r}}$  is determined, such that  $\mathbf{M}_{\mathbf{g}}^{-1}\mathbf{M}_{\mathbf{r}}\mathbf{M}_{\mathbf{f}}$  will register the images. Once  $\mathbf{M}_{\mathbf{r}}$  has been determined,  $\mathbf{M}_{\mathbf{f}}$  can be set to  $\mathbf{M}_{\mathbf{r}}\mathbf{M}_{\mathbf{f}}$ . From there onwards the mapping between the images can be achieved by  $\mathbf{M}_{\mathbf{g}}^{-1}\mathbf{M}_{\mathbf{f}}$ . 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 between  $\mathbf{g}$  and  $\mathbf{h}$  ( $\mathbf{M}_{\mathbf{g}}^{-1}\mathbf{M}_{\mathbf{h}}$ ), but there is also one between  $\mathbf{f}$  and  $\mathbf{h}$  which is simply  $\mathbf{M}_{\mathbf{f}}^{-1}\mathbf{M}_{\mathbf{h}}$  (derived from  $\mathbf{M}_{\mathbf{f}}^{-1}\mathbf{M}_{\mathbf{g}}\mathbf{M}_{\mathbf{g}}^{-1}\mathbf{M}_{\mathbf{h}}$ ).

## 2.4 Optimisation

The objective of optimisation is to determine a set of parameters for which some function of the parameters is minimised (or maximised). One of the simplest cases is determining the optimum parameters for a model in order to minimise the sum of squared differences between the model and a set of real world data ( $\chi^2$ ). Usually there are many parameters in the model, 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  $\chi^2$  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  $\chi^2$  stops decreasing).

The image registration approach described here is essentially an optimisation. One image (the object image) is spatially transformed so that it matches another (the template image), by minimising  $\chi^2$ . 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). The algorithm of choice (Friston *et al.*, 1995b) is one that is similar to *Gauss-Newton* optimisation (see Press *et al.*(1992), Section 15.5 for a fuller explanation of the approach), and it is illustrated here:

Suppose that  $d_i(\mathbf{p})$  is the function describing the difference between the object and template images at voxel  $i$ , when the vector of model parameters have values  $\mathbf{p}$ . For

each voxel ( $i$ ), a first approximation of Taylor's Theorem can be used to estimate the value that this difference will take if the parameters  $\mathbf{p}$  are increased by  $\mathbf{t}$ :

$$d_i(\mathbf{p} + \mathbf{t}) = d_i(\mathbf{p}) + t_1 \frac{\partial d_i(\mathbf{p})}{\partial p_1} + t_2 \frac{\partial d_i(\mathbf{p})}{\partial p_2} \dots$$

From this, a set of simultaneous equations (of the form  $\mathbf{A}\mathbf{x} \simeq \mathbf{b}$ ) can be set up to estimate the values that  $\mathbf{t}$  should take to minimise  $\sum_i d_i(\mathbf{p} + \mathbf{t})^2$ :

$$\begin{pmatrix} -\frac{\partial d_1(\mathbf{p})}{\partial p_1} & -\frac{\partial d_1(\mathbf{p})}{\partial p_2} & \dots \\ -\frac{\partial d_2(\mathbf{p})}{\partial p_1} & -\frac{\partial d_2(\mathbf{p})}{\partial p_2} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \end{pmatrix} \simeq \begin{pmatrix} d_i(\mathbf{p}) \\ d_i(\mathbf{p}) \\ \vdots \end{pmatrix}$$

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

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} + (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (1)$$

where  $\mathbf{A} = \begin{pmatrix} -\frac{\partial d_1(\mathbf{p})}{\partial p_1} & -\frac{\partial d_1(\mathbf{p})}{\partial p_2} & \dots \\ -\frac{\partial d_2(\mathbf{p})}{\partial p_1} & -\frac{\partial d_2(\mathbf{p})}{\partial p_2} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$  and  $\mathbf{b} = \begin{pmatrix} d_i(\mathbf{p}) \\ d_i(\mathbf{p}) \\ \vdots \end{pmatrix}$ .

This process is repeated until  $\chi^2$  can no longer be decreased - or for a fixed number of iterations. There is no guarantee that the best global solution will be reached, since the algorithm can get caught in a local minimum. To reduce this problem, the starting estimates for  $\mathbf{p}$  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 true solution, the registration can continue with less smooth images.

In practice,  $\mathbf{A}^T \mathbf{A}$  and  $\mathbf{A}^T \mathbf{b}$  from Eqn. 1 are 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  $\partial d_i(\mathbf{p})/\partial p_j$  can be rapidly computed from the gradients of the images using the chain rule. These calculations will be illustrated more fully in the next few sections.

## 2.5 Intensity Transformations

The optimisation can be assumed to minimise two sets of parameters: those that describe spatial transformations ( $\mathbf{p}_s$ ), and those for describing intensity transformations ( $\mathbf{p}_t$ ). This means that the difference function can generically be expressed in the form:

$$d_i(\mathbf{p}) = f(\mathbf{s}(\mathbf{x}_i, \mathbf{p}_s)) - t(\mathbf{x}_i, \mathbf{p}_t)$$

where  $\mathbf{f}$  is the object image,  $\mathbf{s}()$  is a vector function describing the spatial transformations based upon parameters  $\mathbf{p}_s$  and  $t()$  is a scalar function describing intensity transformations

based on parameters  $\mathbf{p}_t$ .  $\mathbf{x}_i$  is the  $i$ th coordinates that are sampled. The main sections will simply consider matching one image to a scaled version of another, in order to minimise the sum of squared differences between the images. For this case (assuming that there are 12 parameters describing spatial transformations),  $t(\mathbf{x}_i, \mathbf{p}_t)$  is simply equal to  $p_{13}g(\mathbf{x}_i)$ , where  $p_{13}$  is a simple scaling parameter and  $\mathbf{g}$  is a template image. This is most effective when there is a linear relation between the images. However, the intensities in one image may not vary linearly with the intensities in the other, so it may be more appropriate to match one image to some function of the other image. A simple example of this could be to match image  $\mathbf{f}$  to a scaled version of the square of image  $\mathbf{g}$ . More than one parameter could be used to parameterise the intensity transformation. For example, we could assume some polynomial model for the intensity transformation. In this case,  $t(\mathbf{x}_i, \mathbf{p}_t)$  would equal  $p_{13}g(\mathbf{x}_i) + p_{14}g(\mathbf{x}_i)^2$ , so the and minimised function would have the form:

$$\sum_i (f(\mathbf{x}_i, \mathbf{p}_s) - (p_{13}g(\mathbf{x}_i) + p_{14}g(\mathbf{x}_i)^2))^2$$

Alternatively, the intensities could vary spatially (for example due to inhomogeneities in the MRI scanner). Linear variations can be accounted for by optimising a function of the form:

$$\sum_i (f(\mathbf{x}_i, \mathbf{p}_s) - (p_{13}x_{1i}g(\mathbf{x}_i) + p_{14}x_{2i}g(\mathbf{x}_i) + p_{15}x_{3i}g(\mathbf{x}_i)))^2$$

More complex variations could be included by modulating with other basis functions (such as the DCT basis function set described in Section 6). The examples shown so far have been linear in their parameters describing intensity transformations. A simple example of an intensity transformation that is non-linear would be:

$$\sum_i (f(\mathbf{x}_i, \mathbf{p}_s) - p_{13}g(\mathbf{x}_i)^{p_{14}})^2$$

Another idea is that a given image can be matched not to one reference image, but to a series of images that all conform to the same space. The idea here is that (ignoring the spatial differences) any given image can be expressed as a linear combination of a set of reference images. For example these reference images might include different modalities (e.g., PET, SPECT,  $^{18}\text{F}$ -DOPA,  $^{18}\text{F}$ -deoxy-glucose,  $T_1$ -weighted MRI  $T_2^*$ -weighted MRI .. etc.) or different anatomical tissues (e.g., grey matter, white matter, and CSF segmented from the same  $T_1$ -weighted MRI) or different anatomical regions (e.g., cortical grey matter, sub-cortical grey mater, cerebellum ... etc.) or finally any combination of the above. Any given image, irrespective of its modality could be approximated with a function of these images. A simple example using two images would be:

$$\sum_i (f(\mathbf{M}\mathbf{x}_i) - (p_{13}g_1(\mathbf{x}_i) + p_{14}g_2(\mathbf{x}_i)))^2$$

### 3 Within Modality Image Co-registration

The most common application of within modality co-registration is in motion correction of series of images. It is inevitable that a subject will move slightly during a series of

scans, so prior to performing the statistical tests, it is important that the images are as closely aligned as possible. Subject head movement in the scanner can not be completely eliminated, so motion correction needs to be performed as a preprocessing step on the image data.

Accurate motion correction is especially important for fMRI studies with paradigms where the subject may move in the scanner in a way that is correlated to the different experimental conditions (Hajnal *et al.*, 1994). 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 paradigm may appear as activations. Even after the registration and transformations have been performed, it is likely that the images will still contain artifacts correlated with movement. There are a number of sources of these artifacts, including the approximations used in the interpolation, aliasing effects due to gaps between the slices, ghosts in the images, slices not being acquired simultaneously (so the data no longer obeys the rules of rigid body transformations) and spin excitation history effects. Fortunately (providing that there are enough images in the series), the artifacts can largely be corrected by using an ANCOVA model to remove any signal that is correlated with functions of the movement parameters (Friston *et al.*, 1996). There will be more discussion about this correction in Chapter 9.

A second reason why motion correction is important is that it increases sensitivity. The t-test used by SPM is based on the signal change relative to the residual variance - which 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.

Most current algorithms for movement correction consider the head as a rigid object. In three dimensions, six parameters are needed to define a rigid body transformation (three translations and three rotations). However, we will begin by explaining how the simpler within modality (12 parameter) affine registration can be implemented, before illustrating how the rigid body constraints are incorporated.

### 3.1 Simple Affine Registration

The objective is to fit the image  $\mathbf{f}$  to a template image  $\mathbf{g}$ , using a twelve parameter affine transformation (parameters  $p_1$  to  $p_{12}$ ). The images may be scaled quite differently, so an additional intensity scaling parameter ( $p_{13}$ ) is included in the model.

An affine transformation mapping (via matrix  $\mathbf{M}$ , where the matrix elements are the parameters  $p_1$  to  $p_{12}$ ) from position  $\mathbf{x}$  in one image to position  $\mathbf{y}$  in another is defined by:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} p_1 & p_4 & p_7 & p_{10} \\ p_2 & p_5 & p_8 & p_{11} \\ p_3 & p_6 & p_9 & p_{12} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

We refer to this mapping as  $\mathbf{y} = \mathbf{M}\mathbf{x}$ .

The parameters ( $\mathbf{p}$ ) are optimised by minimising the sum of squared differences between the images according to the algorithm described in Section 2.4 (Eqn. 1). The function that is minimised is:

$$\sum_i (f(\mathbf{M}\mathbf{x}_i) - p_{13}g(\mathbf{x}_i))^2 \quad (2)$$

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

$$b_i = f(\mathbf{M}\mathbf{x}_i) - p_{13}g(\mathbf{x}_i)$$

Matrix  $\mathbf{A}$  is constructed from the negative derivatives. The derivatives are computed as follows:

The rate of change of residual  $i$  with respect to the scaling parameter ( $p_{13}$ ) is simply  $-g(\mathbf{x}_i)$  (the negative intensity of image  $\mathbf{g}$  at  $\mathbf{x}_i$  - the  $i$ th sample position).

The derivatives of the residuals with respect to the spatial transformation parameters ( $p_1$  to  $p_{12}$ ) are obtained by differentiating  $f(\mathbf{M}\mathbf{x}_i) - p_{13}g(\mathbf{x}_i)$  with respect to each parameter ( $p_j$ ) to give  $\partial f(\mathbf{M}\mathbf{x}_i)/\partial p_j$ . The derivatives with respect to the translation parameters are simply the gradient of image  $\mathbf{f}$  in each direction ( $df(\mathbf{y})/dy_1$ ,  $df(\mathbf{y})/dy_2$  and  $df(\mathbf{y})/dy_3$  where  $\mathbf{y}$  is  $\mathbf{M}\mathbf{x}_i$ ). The remaining derivatives are generated from the image gradients using the chain rule:

$$\frac{df(\mathbf{y})}{dp_j} = \frac{df(\mathbf{y})}{dy_1} \frac{dy_1}{dp_j} + \frac{df(\mathbf{y})}{dy_2} \frac{dy_2}{dp_j} + \frac{df(\mathbf{y})}{dy_3} \frac{dy_3}{dp_j}$$

In the above expression, only one of the terms on the right hand side is ever nonzero when the parameters are the elements of the affine transformation matrix, allowing the derivatives to be calculated more rapidly.

## 3.2 Constraining to be Rigid Body

Additional constraints need to be added to convert the algorithm so that it performs a rigid body rather than an affine registration. These are incorporated by re-parameterising from the 12 affine parameters, to the six that are needed to define a rigid body transformation. Matrix  $\mathbf{M}$  is now defined from the six parameters  $\mathbf{q}$  as:

$$\mathbf{M} = \mathbf{M}_f^{-1} \begin{pmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_4) & \sin(q_4) & 0 \\ 0 & -\sin(q_4) & \cos(q_4) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_5) & 0 & \sin(q_5) & 0 \\ 0 & 0 & 0 & 0 \\ -\sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_6) & \sin(q_6) & 0 & 0 \\ -\sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{M}_g$$

where  $\mathbf{M}_f$  and  $\mathbf{M}_g$  are described in Section 2.3.1.

The algorithm is modified to use parameter set  $\mathbf{q}$ , rather than the original  $\mathbf{p}$ , simply by incorporating an additional ( $7 \times 13$ ) matrix  $\mathbf{R}$  such that  $r_{i,j} = dp_j/dq_i$ . Matrix  $\mathbf{R}$  needs to be re-computed in each iteration, but this can be quickly done numerically. The iterative scheme would then become:

$$\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} + (\mathbf{R}^T(\mathbf{A}^T\mathbf{A})\mathbf{R})^{-1}\mathbf{R}(\mathbf{A}^T\mathbf{b})$$

where the braces indicate the most efficient way of performing the computations.

## 4 Between Modality Image Co-registration (and Partitioning)

The 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. This method concentrates on the registration of magnetic resonance (MR) images with positron emission tomography (PET) images, and on co-registering MR images from different scanning sequences. The aim is to co-register images as accurately and quickly as possible, with no manual intervention.

Inter-modality registration of images is less straightforward than that of registering images of the same modality. Two PET images from the same subject generally look similar, so it suffices to find the rigid-body transformation parameters that minimises the sum of squares difference between them. However, for co-registration between modalities there is nothing quite so obvious to minimise. AIR (Woods *et al.*, 1992) is a widely used algorithm for co-registration of PET to MR images, but it has the disadvantage that it depends on pre-processing of the MR images. This normally involves laborious manual editing in order to remove any tissue that is not part of the brain (ie. scalp editing). More recently, the idea of matching images by maximising *mutual information* (MI) is becoming more widespread (Collignon *et al.*, 1995). This elegant approach evolved from methods similar to AIR, and may prove to be more successful than the technique described here.

An alternative method is the manual identification of homologous landmarks in both images. These landmarks are aligned together, thus bringing the images into registration. This is also time-consuming, requires a degree of experience, and can be rather subjective. The method described here requires no pre-processing of the data, or landmark identification, and is still reasonably robust.

This method requires images other than the images that are to be registered ( $\mathbf{f}$  and  $\mathbf{g}$ ). These are template images of the same modalities as  $\mathbf{f}$  and  $\mathbf{g}$  ( $\mathbf{t}_f$  and  $\mathbf{t}_g$ ), and probability images of gray matter, white matter and cerebro-spinal fluid. These probabilistic images will be denoted by the matrix  $\mathbf{B}$  (where each column is a separate image). Images  $\mathbf{t}_f$ ,  $\mathbf{t}_g$  and  $\mathbf{B}$  conform to the same anatomical space.

The between modality co-registration described here is a three step approach:

1. *Determine the affine transformations that map between the images and the templates* by minimisation of the sum of squares differences between  $\mathbf{f}$  and  $\mathbf{t}_f$ , and  $\mathbf{g}$  and  $\mathbf{t}_g$ . These transformations are constrained such that only the parameters that describe the rigid body component are allowed to differ.
2. *Segment or partition the images using the probability images* and a modified *mixture model* algorithm. 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* using the rigid body transformations computed from step 1 as a starting estimate.

For simplicity, we work in a Euclidian space, where measures of distances are expressed in millimeters. To facilitate this, we need to define matrices  $\mathbf{M}_f$ ,  $\mathbf{M}_g$  and  $\mathbf{M}_t$  that map

from the voxel coordinates of images  $\mathbf{f}$ ,  $\mathbf{g}$  and the templates, into their own Euclidian space (see Section 2.3.1). The objective is to determine the affine transformation that maps the coordinate system of  $\mathbf{f}$ , to that of  $\mathbf{g}$ . To accomplish this, we need to find a rigid body transformation matrix  $\mathbf{M}_r$ , such that  $\mathbf{M}_g^{-1}\mathbf{M}_r\mathbf{M}_f$  will co-register the images.

#### 4.1 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 of the same modality using just a twelve (or even nine) 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. We will call these transformation matrices  $\mathbf{M}_{gt}$  and  $\mathbf{M}_{ft}$  respectively. Thus a mapping from  $\mathbf{f}$  to  $\mathbf{g}$  now becomes  $\mathbf{M}_g^{-1}\mathbf{M}_{gt}\mathbf{M}_{ft}^{-1}\mathbf{M}_f$ . However, this affine transformation between  $\mathbf{f}$  and  $\mathbf{g}$  has not been constrained to be rigid body. We modify this simple approach 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 ( $\mathbf{M}_{ta}$ ). ie.  $\mathbf{M}_{gt} = \mathbf{M}_{gr}\mathbf{M}_{ta}$ , and similarly  $\mathbf{M}_{ft} = \mathbf{M}_{fr}\mathbf{M}_{ta}$ . Notice that  $\mathbf{M}_{ta}$  is the same for both  $\mathbf{f}$  and  $\mathbf{g}$ . Now the mapping becomes  $\mathbf{M}_g^{-1}\mathbf{M}_{gr}(\mathbf{M}_{ta}\mathbf{M}_{ta}^{-1})\mathbf{M}_{fr}^{-1}\mathbf{M}_f$ , and is a rigid body transformation.

$$\mathbf{M}_{gr} = \begin{pmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_4) & \sin(q_4) & 0 \\ 0 & -\sin(q_4) & \cos(q_4) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_5) & 0 & \sin(q_5) & 0 \\ 0 & 0 & 0 & 0 \\ -\sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_6) & \sin(q_6) & 0 & 0 \\ -\sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{M}_{fr} = \begin{pmatrix} 1 & 0 & 0 & q_7 \\ 0 & 1 & 0 & q_8 \\ 0 & 0 & 1 & q_9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_{10}) & \sin(q_{10}) & 0 \\ 0 & -\sin(q_{10}) & \cos(q_{10}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_{11}) & 0 & \sin(q_{11}) & 0 \\ 0 & 0 & 0 & 0 \\ -\sin(q_{11}) & 0 & \cos(q_{11}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_{12}) & \sin(q_{12}) & 0 & 0 \\ -\sin(q_{12}) & \cos(q_{12}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{M}_{ta} = \begin{pmatrix} q_{13} & 0 & 0 & 0 \\ 0 & q_{14} & 0 & 0 \\ 0 & 0 & q_{15} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & q_{16} & q_{17} & 0 \\ 0 & 1 & q_{18} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We can now optimise the parameter set  $\mathbf{q} = (q_1 \ q_2 \ \dots)$  in order to determine the transformations that minimise the sum of squares difference between the images and templates. The basic optimisation method has been described in previous sections, and involves generating matrix  $\mathbf{A}^T\mathbf{A}$  and vector  $\mathbf{A}^T\mathbf{b}$  for each iteration, solving the equations and incrementing the parameter estimates.

For the purpose of this optimisation, we define two matrices,  $\mathbf{M}_1 = (\mathbf{M}_t^{-1}\mathbf{M}_{ft}\mathbf{M}_f)^{-1}$ , and  $\mathbf{M}_2 = (\mathbf{M}_t^{-1}\mathbf{M}_{gt}\mathbf{M}_g)^{-1}$ . In the following description of  $\mathbf{A}$  and  $\mathbf{b}$ , we utilise the notation that  $f(\mathbf{x})$  is the intensity of image  $\mathbf{f}$  at position  $\mathbf{x}$ , and similarly for  $g(\mathbf{x})$ ,  $t_f(\mathbf{x})$  and  $t_g(\mathbf{x})$ :

$$\mathbf{A} = \begin{pmatrix} -\frac{df(\mathbf{M}_1\mathbf{x}_1)}{dq_1} & \dots & -\frac{df(\mathbf{M}_1\mathbf{x}_1)}{dq_6} & 0 & \dots & 0 & -\frac{df(\mathbf{M}_1\mathbf{x}_1)}{dq_{13}} & \dots & -\frac{df(\mathbf{M}_1\mathbf{x}_1)}{dq_{18}} & t_f(\mathbf{x}_1) & 0 \\ -\frac{df(\mathbf{M}_1\mathbf{x}_2)}{dq_1} & \dots & -\frac{df(\mathbf{M}_1\mathbf{x}_2)}{dq_6} & 0 & \dots & 0 & -\frac{df(\mathbf{M}_1\mathbf{x}_2)}{dq_{13}} & \dots & -\frac{df(\mathbf{M}_1\mathbf{x}_2)}{dq_{18}} & t_f(\mathbf{x}_2) & 0 \\ \vdots & \vdots \\ 0 & \dots & 0 & -\frac{dg(\mathbf{M}_2\mathbf{x}_1)}{dq_7} & \dots & -\frac{dg(\mathbf{M}_2\mathbf{x}_1)}{dq_{12}} & -\frac{dg(\mathbf{M}_2\mathbf{x}_1)}{dq_{13}} & \dots & -\frac{dg(\mathbf{M}_2\mathbf{x}_1)}{dq_{18}} & 0 & t_g(\mathbf{x}_1) \\ 0 & \dots & 0 & -\frac{dg(\mathbf{M}_2\mathbf{x}_2)}{dq_7} & \dots & -\frac{dg(\mathbf{M}_2\mathbf{x}_2)}{dq_{12}} & -\frac{dg(\mathbf{M}_2\mathbf{x}_2)}{dq_{13}} & \dots & -\frac{dg(\mathbf{M}_2\mathbf{x}_2)}{dq_{18}} & 0 & t_g(\mathbf{x}_2) \\ \vdots & \vdots \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} f(\mathbf{M}_1 \mathbf{x}_1) - q_{19} t_f(\mathbf{x}_1) \\ f(\mathbf{M}_1 \mathbf{x}_2) - q_{19} t_f(\mathbf{x}_2) \\ \vdots \\ g(\mathbf{M}_2 \mathbf{x}_1) - q_{20} t_g(\mathbf{x}_1) \\ g(\mathbf{M}_2 \mathbf{x}_2) - q_{20} t_g(\mathbf{x}_2) \\ \vdots \end{pmatrix}$$

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, we obtain a better solution 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 are derived from the sum of squares difference between the template and object images, obtained from the previous solution of  $\mathbf{q}$ . These are:

$$\frac{I}{\sum_{i=1}^I (f(\mathbf{M}_1 \mathbf{x}_i) - q_{19} t_f(\mathbf{x}_i))^2} \text{ and } \frac{I}{\sum_{i=1}^I (g(\mathbf{M}_2 \mathbf{x}_i) - q_{20} t_g(\mathbf{x}_i))^2}.$$

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

## 4.2 Partitioning the images.

Healthy brain tissue can generally be classified into three broad tissue types on the basis of an MR image. These are gray matter (GM), white matter (WM) and cerebro-spinal fluid (CSF). This classification can be performed manually on a good quality  $T_1$  image, by simply selecting suitable image intensity ranges that encompass most of the voxel intensities of a particular tissue type. However, this manual selection of thresholds is highly subjective.

Many groups have used clustering algorithms to partition MR images into different tissue types, either using images acquired from a single MR sequence, or by combining information from two or more registered images acquired using different scanning sequences (eg. proton-density and  $T_2$ -weighted).

The approach we have adopted here is a modified version of one of these clustering algorithms. The clustering algorithm of choice is the maximum likelihood ‘mixture model’ algorithm (Hartigan, 1975).

We assume that the MR image (or images) consists of a number of distinct tissue types (clusters) from which every voxel has been drawn. The intensities of voxels belonging to each of these clusters conform to a multivariate normal distribution, which can be described by a mean vector, a covariance matrix and the number of voxels belonging to the distribution.

In addition, we have approximate knowledge of the spatial distributions of these clusters, in the form of probability images (provided by the Montreal Neurological Institute (Evans *et al.*, 1992; Evans *et al.*, 1993; Evans *et al.*, 1994)), which have been derived from MR images of a large number of subjects (see Figure 4). The original images were segmented



Figure 4: The prior probability images of GM, WM and CSF (courtesy of the Montreal Neurological Institute).

into binary images of GM, WM and CSF, and all normalised into the same space using a 9 parameter (3 translations, 3 rotations and 3 orthogonal zooms) affine transformation. The probability images are the means of these binary images, so that they contain values in the range of 0 to 1. These images represent the prior probability of a voxel being either GM, WM or CSF after an image has been normalised to the same space using a 9 parameter affine transformation.

The primary difference between the current approach and the pure Mixture Model, is in the update of the belonging probabilities. The assignment of a probability for a voxel is based upon determining the probability of it belonging to cluster  $c_i$  given that it has the intensity  $v$ . This is based upon simple Bayesian statistics:

$$p(c_i|v) = \frac{p(v|c_i)p(c_i)}{\sum_{k=1}^K p(v|c_k)p(c_k)}$$

where  $p(v|c_i)$  is the probability density of the cluster  $c_i$  at value  $v$ , and  $p(c_i)$  is the prior probability of the voxel belonging to cluster  $c_i$ . In the conventional Mixture Model,  $p(c_i)$  is simply  $n_i$  - the number of voxels known to belong to cluster  $c_i$ . In the current implementation,  $p(c_i)$  is based upon knowledge from the *a priori* probability images. It takes the form  $p(c_i) = n_i b_i / \sum b_i$ , where  $b_i$  is the value at the corresponding position of the  $i$ th probability image and  $\sum b_i$  is the integral over this image.

We describe here a simplified version of the algorithm as it would be applied to a single image. We use a 12 parameter affine transformation determined from the previous step to map between the space of the MR image ( $\mathbf{f}$ ), and that of the probability images ( $\mathbf{B}$ ). This allows simple ‘on-the-fly’ sampling of the probability images into the space of the image we wish to partition.

Generally, we use 6 or 7 clusters: one each for GM, WM & CSF, two or three clusters to account for scalp, eyes etc. and a background cluster. Since we have no probability maps for scalp and background, we estimate them by subtracting  $\mathbf{b}_{GM}$ ,  $\mathbf{b}_{WM}$  &  $\mathbf{b}_{CSF}$  from a map of all ones, and divide the results equally between the remaining clusters.

We then assign initial probabilities ( $\mathbf{P}$ ) for each of the  $I$  voxels being drawn from each of the  $K$  clusters. These are based on the *a priori* probability images (ie.  $p_{ik} = b_k(\mathbf{M}_1^{-1}\mathbf{x}_i)$ ). Where identical *a priori* probability maps are used for more than one cluster, the starting estimates are modified slightly by adding random noise.

The following steps (1 to 6) are repeated until convergence (or a prespecified number of iterations) is reached.

1. Compute the number of voxels belonging to each of the  $K$  clusters ( $\mathbf{h}$ ) as:

$$h_k = \sum_{i=1}^I p_{ik} \text{ over } k = 1..K.$$

2. Mean voxel intensities for each cluster ( $\mathbf{v}$ ) are computed. This step effectively produces a weighted mean of the image voxels, where the weights are the current belonging probability estimates:

$$v_k = \frac{\sum_{i=1}^I p_{ik} f(\mathbf{x}_i)}{h_k} \text{ over } k = 1..K.$$

3. Then the variance of each cluster ( $\mathbf{c}$ ) is computed in a similar way to the mean:

$$c_k = \frac{\sum_{i=1}^I p_{ik} (f(\mathbf{x}_i) - v_k)^2}{h_k} \text{ over } k = 1..K.$$

4. Now we have all the parameters that describe the current estimate of the distributions, we have to re-calculate the belonging probabilities ( $\mathbf{P}$ ).

Evaluate the probability density functions for the clusters at each of the voxels:

$$r_{ik} = (2\pi c_k)^{-0.5} \exp\left(\frac{-(f(\mathbf{x}_i) - v_k)^2}{2c_k}\right) \text{ over } k = 1..K \text{ and } i = 1..I.$$

5. Then utilise the prior information ( $\mathbf{B}$ ) (this is the only deviation from the conventional mixture model algorithm that is simply  $q_{ik} = r_{ik} h_k$ ):

$$q_{ik} = r_{ik} \frac{h_k b_k(\mathbf{M}_1^{-1}\mathbf{x}_i)}{\sum_{j=1}^I b_k(\mathbf{M}_1^{-1}\mathbf{x}_j)} \text{ over } k = 1..K \text{ and } i = 1..I.$$

Note that we have extended the mixture model by including an extra term:

$$\frac{b_k(\mathbf{M}_1^{-1}\mathbf{x}_i)}{\sum_{j=1}^I b_k(\mathbf{M}_1^{-1}\mathbf{x}_j)}.$$

This term sums to unity over voxels, and can be thought of as the probability density function of a voxel from cluster  $k$  being found at location  $i$ , irrespective of how many voxels of type  $k$  there are in the brain. Using this term, we can include the prior information, without biasing the overall proportions of different tissue types.

6. And finally normalise the probabilities so that they integrate to unity at each voxel.

$$p_{ik} = \frac{q_{ik}}{\sum_{j=1}^K q_{ij}} \text{ over } k = 1..K \text{ and } i = 1..I.$$



Figure 5: Examples of MR images partitioned into GM, WM and CSF. Top:  $T_2$  weighted image. Bottom:  $T_1$  weighted image.

With each iteration of the algorithm, the parameters describing the distributions ( $\mathbf{v}$ ,  $\mathbf{c}$  &  $\mathbf{h}$ ) move towards a better fit and the belonging probabilities ( $\mathbf{P}$ ) change slightly to reflect the new distributions. The parameters describing the clusters that have corresponding prior probability images tend to converge more rapidly than the other clusters - this is partly due to the better starting estimates. The final values in  $\mathbf{P}$  are in the range of 0 to 1, although most values tend to stabilise very close to one of the two extremes. Examples of MR images classified in this way can be seen in Figure 5.

Strictly speaking, the assumption that multinormal distributions should be used to model MRI intensities is not quite correct. After Fourier reconstruction, the moduli of the complex pixel values are taken, thus rendering any potentially negative values positive. Where the cluster variances are of comparable magnitude to the cluster means, the distribution deviates significantly from normal. This only really applies for the background, where the true mean voxel intensity is zero. The algorithm is modified to account for this discrepancy between the model and reality. For this background cluster, the value of  $v$  is set to zero before the variance  $c$  is computed. Also, because the background cluster is described by only a half Gaussian (and  $h$  represent the integrals of the distributions) it is necessary to double the computed values of  $r$  (step 4 above).

The greatest problem that the technique faces is image non-uniformity. The current algorithm assumes that the voxel values for GM (for example) have the same intensity distribution throughout the image. The non-stationary nature of MR image intensities from some scanners can lead to a significant amount of tissue misclassification.

### 4.3 Co-registering the image partitions.

The previous step produces images of GM, WM and CSF 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. Again, we call these parameters  $\mathbf{q}$ , and define a matrix  $\mathbf{M}_{\mathbf{f}\mathbf{g}}$  based upon these parameters (c.f.  $M_{gr}$  as defined earlier). We define a matrix  $\mathbf{M}$  as  $(\mathbf{M}_{\mathbf{g}}^{-1}\mathbf{M}_{\mathbf{f}\mathbf{g}}\mathbf{M}_{\mathbf{g}\mathbf{t}}^{-1}\mathbf{M}_{\mathbf{f}\mathbf{t}}\mathbf{M}_{\mathbf{f}})^{-1}$ . The way that this matrix has been formulated means that the starting estimates for  $\mathbf{q}$  can all be zero, because it incorporates the results from the first step of the co-registration. Very few iterations are required at this stage to achieve convergence. No scaling parameters are needed, since the probability images derived from  $\mathbf{f}$  have similar intensities to those derived from  $\mathbf{g}$ . The system of equations that we iteratively solve ( $\mathbf{A}\mathbf{x} \simeq \mathbf{b}$ ) to optimise the parameters  $\mathbf{q}$  are as follows (using notation where  $p_{g1}(\mathbf{x}_2)$  means ‘probability of voxel at  $\mathbf{x}_2$  from image  $\mathbf{g}$  belonging to cluster 1’):

$$\mathbf{A} = \begin{pmatrix} -\frac{dp_{f1}(\mathbf{M}\mathbf{x}_1)}{dq_1} & -\frac{dp_{f1}(\mathbf{M}\mathbf{x}_2)}{dq_2} & \dots & p_{g1}(\mathbf{x}_1) & 0 & 0 \\ -\frac{dp_{f1}(\mathbf{M}\mathbf{x}_2)}{dq_1} & -\frac{dp_{f1}(\mathbf{M}\mathbf{x}_2)}{dq_2} & \dots & p_{g1}(\mathbf{x}_2) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -\frac{dp_{f2}(\mathbf{M}\mathbf{x}_1)}{dq_1} & -\frac{dp_{f2}(\mathbf{M}\mathbf{x}_1)}{dq_2} & \dots & 0 & p_{g2}(\mathbf{x}_1) & 0 \\ -\frac{dp_{f2}(\mathbf{M}\mathbf{x}_2)}{dq_1} & -\frac{dp_{f2}(\mathbf{M}\mathbf{x}_2)}{dq_2} & \dots & 0 & p_{g2}(\mathbf{x}_2) & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -\frac{dp_{f3}(\mathbf{M}\mathbf{x}_1)}{dq_1} & -\frac{dp_{f3}(\mathbf{M}\mathbf{x}_1)}{dq_2} & \dots & 0 & 0 & p_{g3}(\mathbf{x}_1) \\ -\frac{dp_{f3}(\mathbf{M}\mathbf{x}_2)}{dq_1} & -\frac{dp_{f3}(\mathbf{M}\mathbf{x}_2)}{dq_2} & \dots & 0 & 0 & p_{g3}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} p_{f1}(\mathbf{M}\mathbf{x}_1) - p_{g1}(\mathbf{x}_1) \\ p_{f1}(\mathbf{M}\mathbf{x}_2) - p_{g1}(\mathbf{x}_2) \\ \vdots \\ p_{f2}(\mathbf{M}\mathbf{x}_1) - p_{g2}(\mathbf{x}_1) \\ p_{f2}(\mathbf{M}\mathbf{x}_2) - p_{g2}(\mathbf{x}_2) \\ \vdots \\ p_{f3}(\mathbf{M}\mathbf{x}_1) - p_{g3}(\mathbf{x}_1) \\ p_{f3}(\mathbf{M}\mathbf{x}_2) - p_{g3}(\mathbf{x}_2) \\ \vdots \end{pmatrix}$$

After this co-registration we have our final solution. It is now possible to map voxel  $\mathbf{x}$  of image  $\mathbf{g}$ , to the corresponding voxel  $\mathbf{M}\mathbf{x}$  of image  $\mathbf{f}$ . Examples of PET-MRI, and T<sub>1</sub>-T<sub>2</sub> co-registration using this approach are illustrated in Figure 6.

#### 4.3.1 An alternative implementation for low resolution images.

Here we briefly describe an approach that may be more appropriate for the registration of SPECT or low resolution PET images to MRI. The tissue classification model described

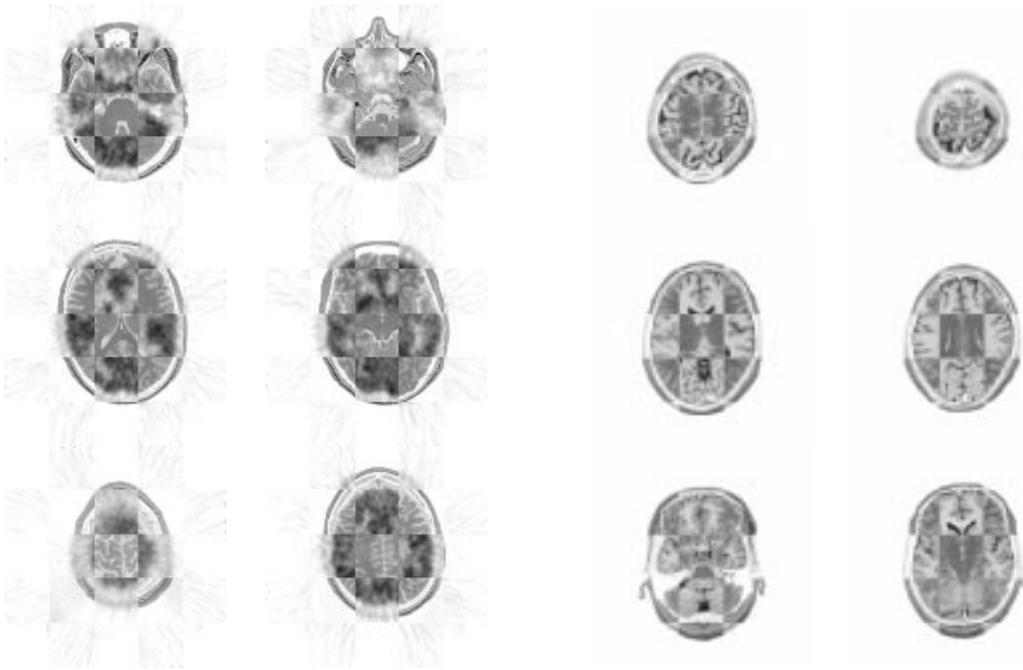


Figure 6: An example of PET-MRI co-registration (Left) and  $T_1$ - $T_2$  co-registration (Right), achieved using the techniques described here.

above is not ideal for partitioning low resolution images. It assumes that each voxel contains tissue from only one of the underlying clusters, whereas in reality, many voxels will contain a mixture of different tissue types (Bullmore *et al.*, 1995; Ashburner *et al.*, 1996).

An alternative is to only partition the MR image as described above, and generate an image from the resulting segments that resembles a PET image. This can be achieved by assigning the gray matter segment a value of 1, white matter a value of about 0.3, and CSF a value of about 0.1, followed by smoothing. It is then possible to apply the within-modality co-registration described in the previous section to co-register the real and ‘fake’ PET images.

#### 4.4 Discussion.

We have described a strategy for the co-registration of brain images from different modalities that is entirely automatic. No manual editing of the images is required in order to remove scalp. Nor does the investigator need to identify any mutual points or features, or even set thresholds for morphological operations like brain segmentation. The only occasional intervention that may be needed is to provide starting estimates to the first step of the procedure. The procedure has so far been successfully applied to the registration of  $T_1$  MRI to PET (blood flow),  $T_1$  to  $T_2$  MRI, and  $T_2$  to PET (in normal subjects).

In addition to providing a method of co-registration, another feature of the current approach is the generation of partitioned (or segmented) images that can be used for voxel based morphometrics (Wright *et al.*, 1995). The incorporation of the prior probabilities

into the clustering algorithm produces a much more robust solution. However, a better result is expected when the method is applied to two (or more) exactly registered images from different scanning sequences. Although the algorithm has only been illustrated for a single image, the principle can be extended such that the classification can be performed using any number of registered images. The mixture model clustering algorithm is described for multi-dimensional input data in Hartigan (1975), although the use of priors is not included in the description.

## 5 Affine Spatial Normalisation

In order to average signals from functional brain images of different subjects, it is necessary to register the images together. This is often done by mapping all the images into the same standard space (Talairach & Tournoux, 1988). Almost all between subject co-registration or spatial normalisation methods for brain images begin with determining the optimal 9 or 12 parameter affine transformation that registers the images together. This step is normally performed automatically by minimising (or maximising) some mutual function of the images. Without constraints and with poor data, the simple parameter optimisation approach can produce some extremely unlikely transformations. For example, when there are only a few transverse slices in the image (spanning the  $X$  and  $Y$  dimensions), it is not possible for the algorithms to determine an accurate zoom in the  $Z$  direction. Any estimate of this value is likely to have very large errors. Previously in this situation, it was better to assign a fixed value for this difficult-to-determine parameter, and simply fit for the remaining ones.

By incorporating prior information into the optimisation procedure, a smooth transition between fixed and fitted parameters can be achieved. When the error for a particular fitted parameter is known to be large, then that parameter will be based more upon the prior information. The approach adopted here is essentially a *maximum a posteriori* (*MAP*) Bayesian approach.

### 5.1 A Bayesian Approach.

Bayes rule is generally expressed in the continuous form:

$$p(a_{\mathbf{p}}|b) = \frac{p(b|a_{\mathbf{p}})p(a_{\mathbf{p}})}{\int_{\mathbf{q}} p(b|a_{\mathbf{q}})p(a_{\mathbf{q}})d\mathbf{q}}$$

where  $p(a_{\mathbf{p}})$  is the prior probability of  $a_{\mathbf{p}}$  being true,  $p(b|a_{\mathbf{p}})$  is the conditional probability that  $b$  is observed given that  $a_{\mathbf{p}}$  is true and  $p(a_{\mathbf{p}}|b)$  is the Bayesian estimate of  $a_{\mathbf{p}}$  being true, given that measurement  $b$  has been made. The *maximum a posteriori* estimate for parameters  $\mathbf{p}$  is the mode of  $p(a_{\mathbf{p}}|b)$ . For our purposes,  $p(a_{\mathbf{p}})$  represents a known prior probability distribution from which the parameters are drawn,  $p(b|a_{\mathbf{p}})$  is the likelihood of obtaining the parameters given the data  $b$  and  $p(a_{\mathbf{p}}|b)$  is the function to be maximised. The optimisation can be simplified by assuming that all probability distributions are multidimensional and normal (multi-normal), and can therefore be described by a mean vector and a covariance matrix.

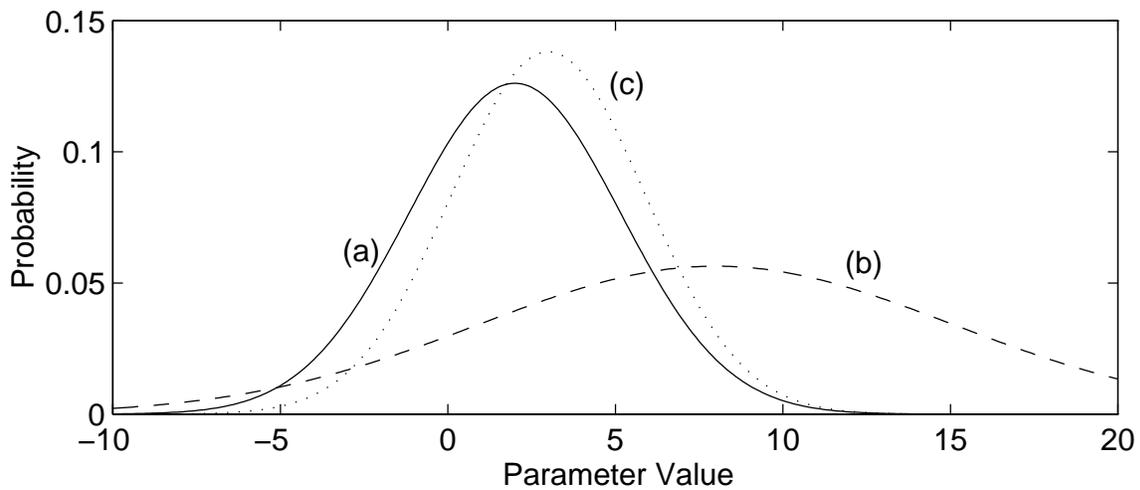


Figure 7: This figure illustrates a hypothetical example with one parameter. The solid Gaussian curve (a) represents the prior probability distribution (p.d.f), and the dashed curve (b) represents a parameter estimate (from fitting to observed data) with its associated certainty. We know that the true parameter was drawn from distribution (a), but we can also estimate it with the certainty described by distribution (b). Without the MAP scheme, we would probably obtain a more precise estimate for the true parameter by taking the most likely *a priori* value, rather than the value obtained from a fit to the data. The dotted line (c) shows the p.d.f that would be obtained from a MAP estimate. It combines previously known information with that from the data to give a more precise estimate.

When close to the minimum, the optimisation becomes almost a linear problem. This allows us to assume that the errors of the fitted parameters ( $\mathbf{p}$ ) can be locally approximated by a multi-normal distribution with covariance matrix  $\mathbf{C}$ . We assume that the true parameters are drawn from a known underlying multi-normal distribution of known mean ( $\mathbf{p}_0$ ) and covariance ( $\mathbf{C}_0$ ). By using the *a priori* probability density function (p.d.f) of the parameters, we can obtain a better estimate of the true parameters by taking a weighted average of  $\mathbf{p}_0$  and  $\mathbf{p}$  (see Figure 7):

$$\mathbf{p}_b = (\mathbf{C}_0^{-1} + \mathbf{C}^{-1})^{-1}(\mathbf{C}_0^{-1}\mathbf{p}_0 + \mathbf{C}^{-1}\mathbf{p}) \quad (3)$$

The estimated covariance matrix of the standard errors for the MAP solution is then:

$$\mathbf{C}_b = (\mathbf{C}_0^{-1} + \mathbf{C}^{-1})^{-1} \quad (4)$$

$\mathbf{p}_b$  and  $\mathbf{C}_b$  are the parameters that describe the multi-normal distribution  $p(a_{\mathbf{p}}|b)$ .

## 5.2 Estimating $\mathbf{C}$ .

In order to employ the Bayesian approach, we need to compute  $\mathbf{C}$ , which is the estimated covariance matrix of the standard errors of the fitted parameters. If the observations are independent, and each has unit standard deviation, then  $\mathbf{C}$  is given by  $(\mathbf{A}^T\mathbf{A})^{-1}$ . In practice, we don't know the standard deviations of the observations, so we assume that it is equal for all observations, and estimate it from the sum of squared differences:

$$\sigma^2 = \sum_{i=1}^I (f(\mathbf{M}\mathbf{x}_i) - p_{13}g(\mathbf{x}_i))^2 \quad (5)$$

This gives a covariance matrix  $(\mathbf{A}^T\mathbf{A})^{-1}\sigma^2/(I - J)$ , where  $I$  refers to the number of sampled locations in the images and  $J$  refers to the number of parameters (13 in this case).

However, complications arise because the images are smooth, resulting in the observations not being independent, and a reduction in the effective number of degrees of freedom (from  $I - J$ ). We correct for the number of degrees of freedom using the principles described by Friston (1995a) (although this approach is not strictly correct (Worsley & Friston, 1995), it gives an estimate that is close enough for our purposes). We can estimate the effective degrees of freedom by assuming that the difference between  $\mathbf{f}$  and  $\mathbf{g}$  approximates a continuous, zero-mean, homogeneous, smoothed *Gaussian random field*. The approximate parameter of the Gaussian point spread function describing the smoothness in direction  $d$  (assuming that the axes of the Gaussian are aligned with the axes of the image coordinate system) can be obtained by (Poline *et al.*, 1995):

$$w_d = \sqrt{\frac{\sigma^2(I - J)}{2\sum_i (\nabla_d(f(\mathbf{M}\mathbf{x}_i) - g(\mathbf{x}_i)))^2}} \quad (6)$$

If the images are sampled on a regular grid where the spacing in each direction is  $s_d$ , the number of effective degrees of freedom ( $\nu$ ) becomes approximately  $(I - J)\prod_d \frac{s_d}{w_d(2\pi)^{1/2}}$ , and the covariance matrix can now be estimated by:

$$\mathbf{C} = (\mathbf{A}^T\mathbf{A})^{-1}\sigma^2/\nu \quad (7)$$

Note that this only applies when  $s_d < w_d(2\pi)^{1/2}$ , otherwise  $\nu = I - J$ .

### 5.3 Estimating $\mathbf{p}_0$ and $\mathbf{C}_0$ .

A suitable *a priori* distribution of the parameters ( $\mathbf{p}_0$  and  $\mathbf{C}_0$ ) was determined from affine transformations estimated from 51 high resolution  $T_1$  weighted brain MR images using basic least squares optimisation algorithm. Each transformation matrix was defined from parameters  $\mathbf{q}$  according to:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_4) & \sin(q_4) & 0 \\ 0 & -\sin(q_4) & \cos(q_4) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(q_5) & 0 & \sin(q_5) & 0 \\ 0 & 0 & 0 & 0 \\ -\sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots$$

$$\dots \times \begin{pmatrix} \cos(q_6) & \sin(q_6) & 0 & 0 \\ -\sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} q_7 & 0 & 0 & 0 \\ 0 & q_8 & 0 & 0 \\ 0 & 0 & q_9 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & q_{10} & q_{11} & 0 \\ 0 & 1 & q_{12} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The results for the translation and rotation parameters ( $q_1$  to  $q_6$ ) can be ignored, since these depend only on the positioning of the subjects in the scanner, and do not reflect variability in head shape and size.

The mean zooms required to fit the individual brains to the space of the template (parameters  $q_7$  to  $q_9$ ) were 1.10, 1.05 and 1.17 in  $X$ ,  $Y$  and  $Z$  respectively, reflecting the fact that the template was larger than the typical head. The covariance matrix was:

$$\begin{pmatrix} 0.00210 & 0.00094 & 0.00134 \\ 0.00094 & 0.00307 & 0.00143 \\ 0.00134 & 0.00143 & 0.00242 \end{pmatrix}$$

giving a correlation coefficient matrix of:

$$\begin{pmatrix} 1.00 & 0.37 & 0.59 \\ 0.37 & 1.00 & 0.52 \\ 0.59 & 0.52 & 1.00 \end{pmatrix}$$

As expected, these parameters are correlated. This allows us to partially predict the optimal zoom in  $Z$  given the zooms in  $X$  and  $Y$ , a fact that is useful for spatially normalising images containing a limited number of transverse slices.

The means of the parameters defining shear were close to zero (-0.0024, 0.0006 and -0.0107 for  $q_{10}$ ,  $q_{11}$  and  $q_{12}$  respectively). The variances of the parameters are 0.000184, 0.000112 and 0.001786, with very little covariance.

## 5.4 Incorporating the Bayesian Approach into the Optimisation.

As mentioned previously, when the parameter estimates are close to the minimum the registration problem is almost linear. Prior to this, the problem is non-linear and covariance matrix  $\mathbf{C}$  no longer directly reflects the certainties of the parameter estimates. However, it does indicate the certainties of the changes made in the parameter estimates at each iteration, so this information can still be incorporated into the iterative optimisation scheme.

By combining Eqns. (1), (3) and (7), we obtain the following scheme:

$$\mathbf{p}_b^{(n+1)} = (\mathbf{C}_0^{-1} + \alpha)^{-1}(\mathbf{C}_0^{-1}\mathbf{p}_0 + \alpha\mathbf{p}_b^{(n)} + \beta) \quad (8)$$

where  $\alpha = \mathbf{A}^T\mathbf{A}\nu/\sigma^2$  and  $\beta = \mathbf{A}^T\mathbf{b}\nu/\sigma^2$ .

Another way of thinking about this optimisation scheme, is that two criteria are simultaneously being minimised. The first is the sum of squares difference between the images, and the second is a scaled distance squared between the parameters and their known expectation.

### 5.4.1 Stopping Criterion.

The optimal solution is no longer that which minimises the sum of squares of the residuals, so the rate of change of  $\sigma^2$  is not the best indication of when the optimisation has converged. The objective of the optimisation is to obtain a fit with the smallest errors. These errors are described by the covariance matrix of the parameter estimates, which in the case of this optimisation scheme is  $(\alpha + \mathbf{C}_0^{-1})^{-1}$ . The ‘tightness’ of the fit is reflected in the determinant of this matrix, so the optimal solution should be achieved when the determinant is minimised. In practice we look at the rate of change of the log of the determinant.

## 6 Non-linear Spatial Normalisation

Statistical Parametric Mapping using positron emission tomography (PET) or functional magnetic resonance images (fMRI) necessitates the transformation of images from several subjects into the same anatomical space. The basic idea is to use a target (or template) image to define the standard space into which the different subjects are warped. By using a template which conforms to the space of a standard coordinate system, such as that defined by Talairach and Tournoux (1988), it is possible to report anatomical positions in terms of Cartesian coordinates, relative to some reference.

There are a number of approaches to non-linear spatial normalisation. Some of these are interactive, requiring the user to select homologous landmarks in the object and target images to be co-registered using non-linear warps or deformations. The most notable of these are the thin plate spline algorithms (Bookstein, 1989). However, the interactive identification of landmarks is time consuming, and also rather subjective.

Other methods attempt to perform spatial normalisation in an automatic manner. There is a potentially enormous number of parameters that could be solved for in spatial normalisation problems (ie. the problem is very high dimensional). The forms of spatial normalisation tend to differ in how they cope with the large number of parameters required to define the transformation. Some have abandoned the conventional optimisation approach, and use viscous fluid models (Christensen *et al.*, 1993; Christensen *et al.*, 1996). The major advantage of these methods is that they ensure a one-to-one mapping in the allowed spatial transformations. Others adopt a multi-resolution approach whereby only a few of the parameters are determined at any one time (Collins *et al.*, 1994b). Usually, the entire volume is used to determine parameters that describe overall low frequency deformations. The volume is then subdivided, and slightly higher frequency deformations are found for each subvolume. This continues until the desired deformation precision is achieved.

Another approach is to reduce the number of parameters that model the deformations. This is often done by describing the deformation by a linear combination of basis functions. This section describes one such approach.

The deformations required to transform images to the same space are not clearly defined. Unlike rigid body transformations, where the constraints are explicit, those for non-linear warping are more arbitrary. Different subjects have different patterns of gyral convolutions, so there is not necessarily a single best transformation from one space to another. Even if gyral anatomy can be matched exactly, this is no guarantee that areas of functional specialisation will be matched in a homologous way. For the purpose of averaging signals from functional images of different subjects, very high resolution spatial normalisation may be unnecessary or unrealistic.

## 6.1 A Basis Function Approach

The model for defining the non-linear warping uses deformations that consist of a linear combination of basis functions. So, the transformation from coordinates  $\mathbf{x}$ , to coordinates  $\mathbf{y}$  is:

$$y_1 = x_1 + \sum_j t_{j1} b_{j1}(\mathbf{x})$$

$$y_2 = x_2 + \sum_j t_{j2} b_{j2}(\mathbf{x})$$

$$y_3 = x_3 + \sum_j t_{j3} b_{j3}(\mathbf{x})$$

where  $t_{jd}$  is the  $i$ th coefficient for dimension  $d$ , and  $b_{jd}(\mathbf{x})$  is the  $j$ th basis function at position  $\mathbf{x}$  for dimension  $d$ . The basis functions used are those of the three dimensional discrete cosine transform (DCT), because they have useful properties (which will be explained later). The two dimensional DCT basis functions are shown in Figure 8, and a schematic application of a deformation is shown for a two dimensional example in Figure 9.

Again, the optimisation involves minimising the sum of squared differences between the object image ( $\mathbf{f}$ ) and a template image ( $\mathbf{g}$ ). The images may be scaled differently, so

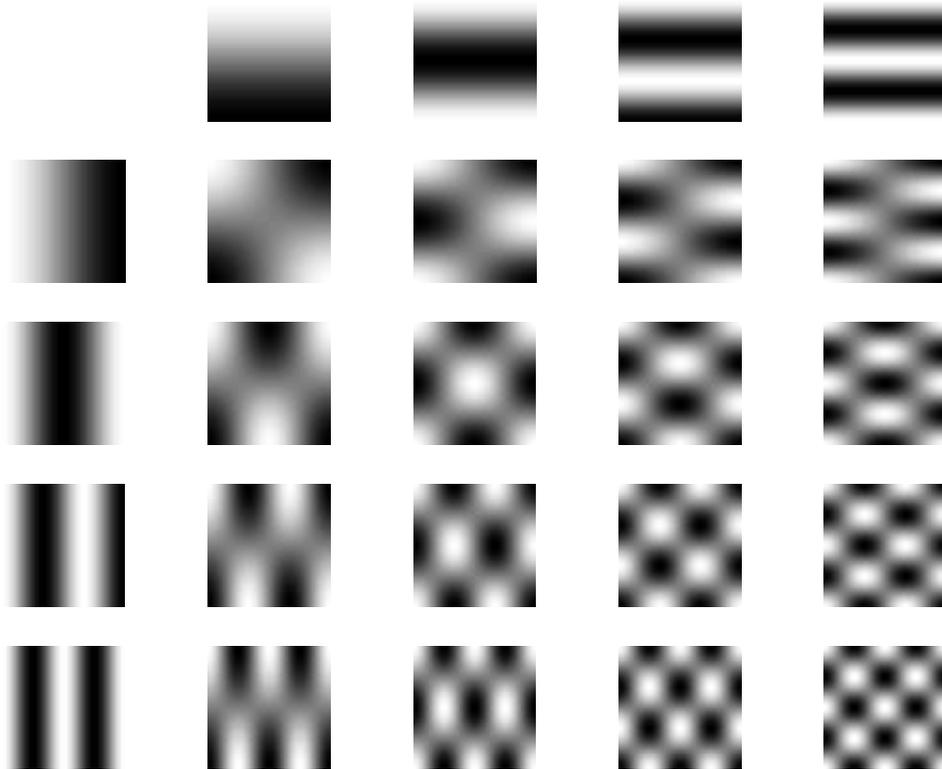


Figure 8: The lowest frequency basis functions of a two dimensional Discrete Cosine Transform basis functions.

an additional parameter ( $u$ ) is needed to accommodate this difference. The optimised function is:

$$\sum_i f(\mathbf{y}_i) - ug(\mathbf{x}_i)$$

The approach described in Section 2.4 is used to optimise the parameters  $\mathbf{t}_1$ ,  $\mathbf{t}_2$ ,  $\mathbf{t}_3$  and  $u$ . This requires the derivatives of the function  $f(\mathbf{y}_i) - ug(\mathbf{x}_i)$  with respect to each parameter, and these can be obtained using the chain rule:

$$\frac{df(\mathbf{y})}{dt_{j1}} = \frac{df(\mathbf{y})}{dy_1} \frac{dy_1}{dt_{j1}}$$

$$\frac{df(\mathbf{y})}{dt_{j2}} = \frac{df(\mathbf{y})}{dy_2} \frac{dy_2}{dt_{j2}}$$

$$\frac{df(\mathbf{y})}{dt_{j3}} = \frac{df(\mathbf{y})}{dy_3} \frac{dy_3}{dt_{j3}}$$

In these expressions,  $df(\mathbf{y})/dy_d$  is simply the derivative in dimension  $d$  of image  $\mathbf{f}$ , and  $dy_d/dt_{di}$  simply evaluates to  $b_{jd}(\mathbf{x})$ .

## 6.2 A Fast Algorithm

In this section, a slightly different mathematical notation is used in order to illustrate (using matrix terminology) how the computations are actually performed. The illustra-

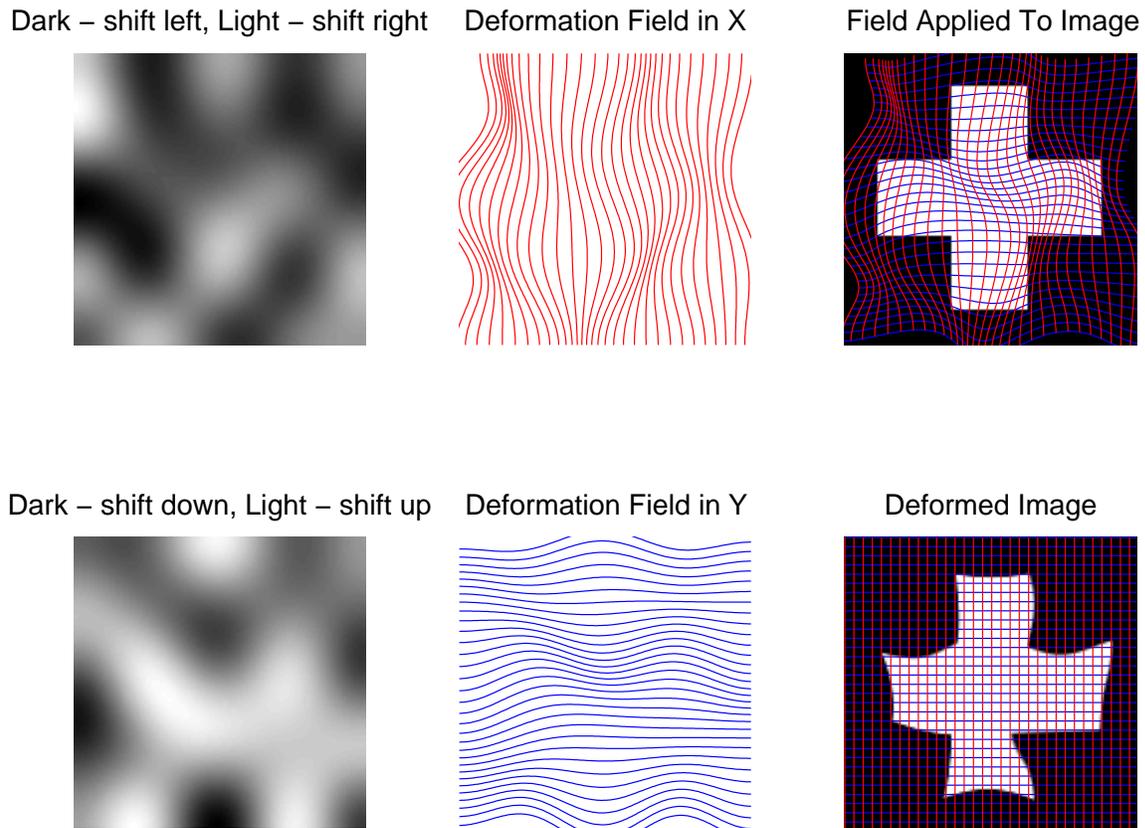


Figure 9: For the two dimensional case, the deformation field consists of two scalar fields. One for horizontal deformations, and the other for vertical deformations. The images on the left show the deformation fields as a linear combination of the basis images (see Figure 8). The center column shows the deformations in a more intuitive sense. The deformation field is applied by overlaying it on the object image, and re-sampling (right).

tion is in two dimensions and it is left to the reader to generalise to three dimensions. The images  $\mathbf{f}$  and  $\mathbf{g}$  are considered as matrices  $\mathbf{F}$  and  $\mathbf{G}$  respectively. For matrix  $\mathbf{F}$ , the value of the element at position  $m,n$  is denoted by  $f_{m,n}$ . Row  $m$  of the same matrix will be denoted by  $\mathbf{f}_{m,:}$ , and column  $n$  by  $\mathbf{f}_{:,n}$ . The transform coefficients are also treated as matrices  $\mathbf{T}_x$  and  $\mathbf{T}_y$ . The basis functions used by the algorithm can be generated from a separable form from matrices  $\mathbf{B}_x$  and  $\mathbf{B}_y$ , such that the deformation fields can be rapidly constructed by computing  $\mathbf{B}_x \mathbf{T}_x \mathbf{B}_y^T$  and  $\mathbf{B}_x \mathbf{T}_y \mathbf{B}_y^T$ .

The basis functions of choice are the lowest frequency components of the two dimensional discrete cosine transform. This transform was chosen because the two dimensional DCT is separable, it is a real transform (eliminating the need for complex arithmetic) and because it's lowest frequencies give excellent energy compaction for smooth functions (Jain, 1989). In one dimension, the DCT of a function is generated by multiplication with the matrix  $\mathbf{B}^T$ , where the elements of  $\mathbf{B}$  are defined by:

$$b_{m,1} = \frac{1}{\sqrt{M}} \quad m=1..M$$

$$b_{m,i} = \sqrt{\frac{2}{M}} \cdot \cos\left(\frac{\pi \cdot (2 \cdot m - 1) \cdot (i - 1)}{(2 \cdot M)}\right) \quad m=1..M, i=2..I$$

Between each iteration, the image  $\mathbf{F}$  is resampled according to the latest parameter estimates. The derivatives of  $\mathbf{F}$  are also resampled to give  $\nabla_x \mathbf{F}$  and  $\nabla_y \mathbf{F}$ . The algorithm for generating  $\mathbf{A}^T \mathbf{A}$  and  $\mathbf{A}^T \mathbf{b}$  ( $\alpha$  and  $\beta$ ) for each iteration is then:

$$\alpha = (\mathbf{0})$$

$$\beta = (\mathbf{0})$$

for  $j = 1 \dots J$

$$\mathbf{C} = \mathbf{b}_{y,j,:}^T \mathbf{b}_{y,j,:}$$

$$\mathbf{E}_x = \text{diag}(-\nabla_x \mathbf{f}_{:,j}) \mathbf{B}_x$$

$$\mathbf{E}_y = \text{diag}(-\nabla_y \mathbf{f}_{:,j}) \mathbf{B}_x$$

$$\alpha = \alpha + \begin{pmatrix} \mathbf{C} \otimes (\mathbf{E}_x^T \mathbf{E}_x) & \mathbf{C} \otimes (\mathbf{E}_x^T \mathbf{E}_y) & \mathbf{b}_{y,j,:}^T \otimes (\mathbf{E}_x^T \mathbf{g}_{:,j}) \\ (\mathbf{C} \otimes (\mathbf{E}_x^T \mathbf{E}_y))^T & \mathbf{C} \otimes (\mathbf{E}_y^T \mathbf{E}_y) & \mathbf{b}_{y,j,:}^T \otimes (\mathbf{E}_y^T \mathbf{g}_{:,j}) \\ (\mathbf{b}_{y,j,:}^T \otimes (\mathbf{E}_x^T \mathbf{g}_{:,j}))^T & (\mathbf{b}_{y,j,:}^T \otimes (\mathbf{E}_x^T \mathbf{g}_{:,j}))^T & \mathbf{g}_{:,j}^T \mathbf{g}_{:,j} \end{pmatrix}$$

$$\beta = \beta + \begin{pmatrix} \mathbf{b}_{y,j,:}^T \otimes (\mathbf{E}_x^T \mathbf{f}_{:,j}) \\ \mathbf{b}_{y,j,:}^T \otimes (\mathbf{E}_y^T \mathbf{f}_{:,j}) \\ \mathbf{g}_{:,j}^T \mathbf{f}_{:,j} \end{pmatrix}$$

end

In the above algorithm, the symbol ' $\otimes$ ' refers to the *Kronecker tensor product*. If  $\mathbf{A}$  is a matrix of order  $M \times N$ , and  $\mathbf{B}$  is a second matrix, then:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11} \mathbf{B} & \dots & a_{1N} \mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{M1} \mathbf{B} & \dots & a_{MN} \mathbf{B} \end{pmatrix}$$

The notation  $\text{diag}(-\nabla_x \mathbf{f}_{:,j}) \mathbf{B}_x$  simply means multiplying each element of row  $i$  of  $\mathbf{B}_x$  by  $-\nabla_x \mathbf{f}_{i,j}$ .

This rather cumbersome looking algorithm is used since it utilises some of the useful properties of Kronecker tensor products. This is especially important when the algorithm is implemented in three dimensions. The performance enhancement results from a reordering of a set of operations like  $(\mathbf{B}_z \otimes \mathbf{B}_y \otimes \mathbf{B}_x)^T (\mathbf{B}_z \otimes \mathbf{B}_y \otimes \mathbf{B}_x)$ , to the equivalent  $(\mathbf{B}_z^T \mathbf{B}_z) \otimes (\mathbf{B}_y^T \mathbf{B}_y) \otimes (\mathbf{B}_x^T \mathbf{B}_x)$ . Assuming that the matrices  $\mathbf{B}_z$ ,  $\mathbf{B}_y$  and  $\mathbf{B}_x$  all have order  $M \times N$ , then the number of floating point operations is reduced from  $M^3 N^3 (N^3 + 2)$  to approximately  $3M(N^2 + N) + N^6$ . If  $M$  equals 32, and  $N$  equals 4, we expect a performance increase of about a factor of 23,000.

### 6.3 Regularisation - a Bayesian-like approach

As the algorithm stands, it is possible to introduce unnecessary deformations that only reduce the residual sum of squares by a tiny amount. In this section we describe a form of regularisation for biasing the deformations to be smooth, and so improve stability. The principles behind the regularisation are Bayesian, and are essentially the same as those described in Section 5. A Bayesian approach to non-linear image registration is nothing new. The incorporation of prior knowledge about the properties of the allowed warps is fundamental to all successful non-linear registration approaches. Gee *et al.*(1995) have described one Bayesian approach to non-linear image registration.

The objective of spatial normalisation is to warp the images such that homologous regions of different brains are moved as close together as possible. A large number of parameters are required to encompass the range of possible non-linear warps. With many parameters relative to the number of independent observations, the errors associated with the fit are likely to be very large. The use of constraints (such as preserving a one-to-one mapping between image and template) can reduce these errors, but they still remain considerable. For this purpose, the simple minimisation of differences between the images is not sufficient. Although the normalised images may appear similar to each other, the data may in-fact have been ‘over-fitted’, resulting in truly homologous regions being moved further apart. Other researchers circumvent this over-fitting problem by restricting their spatial normalisation to just an affine transformation. A properly implemented Bayesian approach should attempt to reach an optimum compromise between these two extremes. Although the incorporation of an optimally applied MAP approach into non-linear registration has the effect of biasing the resulting deformations to be smoother than the true deformations, it is envisaged that homologous voxels should be registered more closely than for unconstrained deformations.

This regularisation is achieved by minimising the sum of squares difference between the template and the warped image, while simultaneously minimising the sum of squares of the derivatives of the deformation field. If we assume linearity, in two dimensions this can be expressed as minimising an expression of the form  $|\mathbf{A}\mathbf{p} - \mathbf{b}|^2 + \lambda(|\mathbf{D}_x\mathbf{p}|^2 + |\mathbf{D}_y\mathbf{p}|^2)$ , where  $\mathbf{D}_x\mathbf{p}$  and  $\mathbf{D}_y\mathbf{p}$  represent the derivatives of the deformation field. The parameter  $\lambda$  (ranging from 0 to infinity) simply describes how much emphasis should be placed upon the smoothness of the final solution. From this, we can derive an iterative scheme similar to that shown in Eqn. 8:

$$\mathbf{p}^{(n+1)} = (\alpha + \lambda(\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y))^{-1}(\alpha \mathbf{p}^{(n)} + \beta).$$

The diagonal matrix  $\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y$  can be readily computed, and is equivalent to  $\mathbf{C}_0^{-1}$  from Eqn. 8. Brain lengths vary with a standard deviation of about 5% of the mean, and may be appropriate to assume that there is roughly the same variability in the lengths of the different brain sub-structures. The relative sizes of voxels before and after spatial normalisation is reflected in the derivatives of the fields that describe the deformation. Therefore, the optimum value for  $\lambda$  may be one that reflects a prior distribution of these derivatives with a standard deviation of about 0.05.

If the true prior distribution of the parameters is known (derived from a large number of subjects), then the matrix  $(\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y)$  could be replaced by the inverse of the covariance matrix that describes this distribution. This approach would have the advantage that the resulting deformations are more typically “brain like”, and so increases the face validity of the approach.

## 6.4 Discussion

The criteria for ‘good’ spatial transformations can be framed in terms of validity, reliability and computational efficiency. The validity of a particular transformation device is not easy to define or measure and indeed varies with the application. For example a rigid body transformation may be perfectly valid for realignment but not for spatial normalisation of an arbitrary brain into a standard stereotactic space. Generally the sorts of validity that are important in spatial transformations can be divided into (i) *Face validity*, established by demonstrating the transformation does what it is supposed to and (ii) *Construct validity*, assessed by comparison with other techniques or constructs. In functional mapping face validity is a complex issue. At first glance, face validity might be equated with the co-registration of anatomical homologues in two images. This would be complete and appropriate if the biological question referred to structural differences or modes of variation. In other circumstances however this definition of face validity is not appropriate. For example the purpose of spatial normalisation (either within or between subjects) in functional mapping studies is to maximise the sensitivity to neurophysiological change elicited by experimental manipulation of sensorimotor or cognitive state. In this case a better definition of a valid normalisation is that which maximises condition-dependent effects with respect to error (and if relevant inter-subject) effects. This will probably be effected when functional anatomy is congruent. This may or may not be the same as registering structural anatomy.

The method described here does not have the potential precision of some other methods for computing non-linear deformations, since the deformations are only defined by a few hundred parameters. However, it may be meaningless to attempt an exact match between brains beyond a certain resolution. There is not a one-to-one relationship between the cortical structures of one brain and those of another, so any method that claims to match brains exactly must be folding the brain to create sulci and gyri that do not really exist.

The current method is relatively fast (takes in the order of 30 seconds per iteration). The speed is partly a result of the small number of parameters involved, and the simple optimisation algorithm that assumes an almost quadratic error surface. Because the images are first matched using a simple affine transformation, there is less ‘work’ for the algorithm to do, and a good registration can be achieved with only a few iterations

(about 10).

When higher spatial frequency deformations are to be fitted, more DCT coefficients are required to describe the deformations. There are practical problems that occur when more than about the  $8 \times 8 \times 8$  lowest frequency DCT components are used. One of these is the problem of storing and inverting the curvature matrix ( $\mathbf{A}^T \mathbf{A}$ ). Even with deformations limited to  $8 \times 8 \times 8$  coefficients, there are at least 1537 unknown parameters, requiring a curvature matrix of about 18Mbytes (using double precision floating point arithmetic). An alternative optimisation method (which does not require this storage) is needed when more parameters are to be estimated. One possible approach is to substitute the Gauss-Newton optimisation for a conjugate gradient method (Press *et al.*, 1992).

A second problem with attempting to fit higher spatial frequencies (especially with very little regularisation), is one of stability. With too many parameters, the lack of hard constraints can result in extremely unlikely deformations. Part of the stability problem is due to there being nothing to constrain the deformation fields to be diffeomorphic (ie. a unique one-to-one correspondence from one space to the other - see (Christensen *et al.*, 1996)). The use of a limited number of smooth basis functions, and the regularisation technique described in this paper only reduces the likelihood of the diffeomorphism constraint being broken. Although this approach ensures that each point in the template's space maps to only one point in the space of the object, the reverse is not true, and it is possible to produce deformations that loop back on themselves.

In order to satisfy the diffeomorphism constraint, future work may involve restricting the determinants of the Jacobians at each point of the deformation field to be greater than zero. The current method of regularisation assumes a prior distribution for these determinants that allows them to be negative. If the prior distribution were assumed to be log-normal, then the non-negativity constraint on the determinants would be enforced. The choice of a log-normal distribution is because the determinants of the Jacobians represent the relative volumes of the warped and unwarped voxels, and the likelihood of a voxel doubling in volume should be the same as the likelihood of its volume being halved.

Optimisation problems are best solved with smooth functions. For this particular case, the template and object images are smoothed (to about 8mm FWHM) to facilitate the optimisation. Not only does using smooth images facilitate a more rapid solution to the problem, but it also reduces the number of possible local minima. However, it does have the disadvantage that some fine grain information is lost. An approach that successively optimises the problem at different resolutions may be more appropriate. This would involve normalising initially at a low resolution to determine the lowest frequency deformations, and gradually increasing the resolution in order to determine the finer deformations. This is an approach that has been adopted by a number of investigators (e.g. Collins *et al.*). However, even using a multiscale approach, there is no guarantee of obtaining the global minimum solution. A further complication arises from the fact that there is no one-to-one match between the small structures (especially gyral and sulcal patterns) of any two brains. Even identical twins have different patterns of cortical folding (Bartley *et al.*, 1997). This means that it is not possible to obtain an objective high frequency match however good the algorithm is. For the purposes of functional imaging, it is probably best to accept that there is a lower bound in terms of how accurately it is possible to register cortical areas together. If one accepts this, then it

seems logical to opt for a smooth solution, and use only a small number of parameters to define the deformations.

## 7 Summary

Prior to any statistical analysis within the *SPM* package, it is important that all the functional images (PET or fMRI) from each subject are aligned together. This step is performed by determining a rigid body transformation for each of the images that registers them to the first in the series. This step is a within modality procedure, and results in an optimisation of the parameters by minimising the residual sum of squares.

Often, it is desirable to register a structural image to the functional image series. Again, this is a rigid body registration, but because the structural image is acquired in a different modality to the functional images, the registration can not simply be performed by minimising the residual sum of squares. Within *SPM*, these between modality registrations are performed by first partitioning the images into gray and white matter and then simultaneously registering the partitions together.

Images from several subjects can be analyzed together by first normalising them all to the same space. In order to facilitate reporting of significant activations by their location within a standard coordinate system, this space is usually that described by Talairach and Tournoux (1988). Brains vary in shape and size so more parameters are needed to describe the spatial transformations. The spatial normalisation usually begins by determining the optimum 12 parameter affine transformation to register the brain with a template image. The template image is of the same modality as the image to be registered, so the optimisation is simply done by minimising the residual sum of squares. This 12 parameter transformation corrects for the variation in position and size of the image, before more subtle differences are corrected by a non-linear registration. In order to reduce the number of parameters to be fitted, only smoothly varying deformations are determined by the non-linear registration. These deformations are modeled by a linear combination of smooth basis functions, and a fast optimisation method has been developed to determine the best coefficients for each of the basis functions. Once the transformation parameters have been determined from one image, they can be applied to any other image that is in register with it.

# References

- Alpert, N. M., Bradshaw, J. F., Kennedy, D., & Coreia, J. A. 1990. The principal axis transformation - a method for image registration. *J. nucl. med.*, **31**, 1717–1722.
- Ashburner, J., Haslam, J., Taylor, C., Cunningham, V. J., & Jones, T. 1996. A cluster analysis approach for the characterization of dynamic PET data. *Chap. 59, pages 301–306 of: Quantification of brain function using PET.*
- Bartley, A. J., Jones, D. W., & Weinberger, D. R. 1997. Genetic variability of human brain size and cortical gyral patterns. *Brain*, **120**, 257–269.
- Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE trans pattern anal machine intelligence*, **11(6)**, 567–585.
- Bullmore, E., Brammer, M., Rouleau, G., Everitt, B., Simmons, A., Sharma, T., Frangou, S., Murray, R., & Dunn, G. 1995. Computerized brain tissue classification of magnetic resonance images: A new approach to the problem of partial volume artifact. *NeuroImage*, **2**, 133–147.
- Christensen, G. E., Rabbitt, R. D., & Miller, M. I. 1993. 3D brain mapping using using a deformable neuroanatomy. *Physics in medicine and biology*, **39**, 609–618.
- Christensen, G. E., Rabbitt, R. D., & Miller, M. I. 1996. Deformable templates using large deformation kinematics. *IEEE transactions on image processing*, **5**, 1435–1447.
- Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., & Marchal, G. 1995. Automated multi-modality image registration based on information theory. *Pages 263–274 of: Information processing in medical imaging.*
- Collins, D. L., Peters, T. M., & Evans, A. C. 1994a. An automated 3D non-linear image deformation procedure for determination of gross morphometric variability in human brain. *Proc. conference on visualisation in biomedical computing*, 180–190.
- Collins, D. L., Neelin, P., Peters, T. M., & Evans, A. C. 1994b. Automatic 3D intersubject registration of MR volumetric data in standardized taliarach space. *J. comput. assist. tomogr.*, **18**, 192–205.
- Eddy, W. F., Fitzgerald, M., & Noll, D. C. 1996. Improved image registration by using Fourier interpolation. *Magnetic resonance in medicine*, **36**, 923–931.
- Evans, A. C., Collins, D. L., & Milner, B. 1992. An MRI-based stereotactic atlas from 250 young normal subjects. *Soc. neurosci. abstr.*, **18**, 408.
- Evans, A. C., Collins, D. L., Mills, S. R., Brown, E. D., Kelly, R. L., & Peters, T. M. 1993. 3D statistical neuroanatomical models from 305 MRI volumes. *Pages 1813–1817 of: Proc. IEEE-nuclear science symposium and medical imaging conference.*
- Evans, A. C., Kamber, M., Collins, D. L., & Macdonald, D. 1994. An MRI-based probabilistic atlas of neuroanatomy. *Pages 263–274 of: Shorvon, S., Fish, D., Andermann, F., Bydder, G. M., & H, Stefan (eds), Magnetic resonance scanning and epilepsy. NATO ASI Series A, Life Sciences, vol. 264. Plenum Press.*
- Fox, P. T. 1995. Spatial normalization origins: Objectives, applications, and alternatives. *Human brain mapping*, **3**, 161–164.

- Friston, K. J., Frith, C. D., Liddle, P. F., & Frackowiak, R. S. J. 1991. Plastic transformation of PET images. *J. comput. assist. tomogr.*, **15**, 634–639.
- Friston, K. J., Holmes, A. P., Poline, J.-B., Grasby, P. J., Williams, S. C. R., Frackowiak, R. S. J., & Turner, R. 1995a. Analysis of fMRI time series revisited. *NeuroImage*, **2**, 45–53.
- Friston, K. J., Ashburner, J., Frith, C. D., Poline, J.-B., Heather, J. D., & Frackowiak, R. S. J. 1995b. Spatial registration and normalization of images. *Human brain mapping*, **2**, 165–189.
- Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S. J., & Turner, R. 1996. Movement-related effects in fMRI time-series. *Magnetic resonance in medicine*, **35**, 346–355.
- Gee, J. C., Briquer, L. Le, & Barillot, C. 1995. Probabilistic matching of brain images. *Pages 113–125 of: Information processing in medical imaging*.
- Hajnal, J. V., Mayers, R., Oatridge, A., Schwieso, J. E., Young, J. R., & Bydder, G. M. 1994. Artifacts due to stimulus correlated motion in functional imaging of the brain. *Magnetic resonance in medicine*, **31**, 289–291.
- Hartigan, J. A. 1975. *Clustering algorithms*. New York: John Wiley & Sons, Inc. Pages 113–129.
- Jain, A. K. 1989. *Fundamentals of digital image processing*. Prentice-Hall.
- Kosugi, Y., Sase, M., Kuwatani, H., Kinoshita, N., Momose, T., Nishikawa, J., & Watanabe, T. 1993. Neural network mapping for nonlinear stereotactic normalisation of brain MR images. *J. comput. assist. tomogr.*, **17**, 455–460.
- Lange, N. 1994. Some computational and statistical tools for paired comparisons of digital images. *Statistical methods in medical research*, **3**, 23–40.
- Pelizzari, C. A., Chen, G. T. Y., Spelbring, D. R., Weichselbaum, R. R., & Chen, C. T. 1988. Accurate three-dimensional registration of CT, PET and MR images of the brain. *J. comput. assist. tomogr.*, **13**, 20–26.
- Poline, J.-B., Friston, K. J., Worsley, K. J., & Frackowiak, R. S. J. 1995. Estimating smoothness in statistical parametric maps: Confidence intervals on  $p$ -values. *J. comput. assist. tomogr.*, **19**(5), 788–796.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992. *Numerical recipes in C (second edition)*. Cambridge: Cambridge.
- Talairach, J., & Tournoux. 1988. *Coplanar stereotaxic atlas of the human brain*. New York: Thieme Medical.
- Woods, R. P., Cherry, S. R., & Mazziotta, J. C. 1992. Rapid automated algorithm for aligning and reslicing PET images. *J. comput. assist. tomogr.*, **16**, 620–633.
- Worsley, K. J., & Friston, K. J. 1995. Analysis of fMRI time-series revisited - again. *NeuroImage*, **2**, 173–181.
- Wright, I. C., McGuire, P. K., Poline, J.-B., Travere, J. M., Murray, R. M., Frith, C. D., Frackowiak, R. S. J., & Friston, K. J. 1995. A voxel-based method for the statistical analysis of gray and white matter density applied to schizophrenia. *NeuroImage*, **2**, 244–252.