# Global Intensity Normalization induces Correlation in fMRI

Kevin K. Liu[1] and Daniel B. Rowe[1,2]
[1]Department of Mathematics, Statistics and Computer Science, Marquette University, Milwaukee, WI, USA
[2]Department of Biophysics, Medical College of Wisconsin, Milwaukee, WI, USA

## Abstract

Subject brains vary in both size and physical features. In functional magnetic resonance imaging (fMRI), the most intuitive approach to correcting discrepancies between scans are processing with spatial and temporal filtering, whether it be a high or low filter, depending upon your intentions. A less obvious approach is global intensity normalization. As mentioned before, due to the uniqueness of each individual, the mean intensity of an entire data set will vary across individuals and scans. The goal is to retain a constant signal mean across each independent scan. Although the purpose of intensity normalization is to adjust for a shortfall in the data, a side effect is that it may induce a correlation of no biological origin, leading to inaccurate interpretations of the data.

### 1.1 Introduction

In order to account for discrepancies between each individual's scans, the process of global intensity normalization registers each image with a template image. In doing so, each individual's scans is then aligned into the same coordinate space. The alignment of multiple subject's scans to a template allows for group statistics and comparison between subjects despite the discrepancies between them. This makes Spatial Normalization a very valuable and common part of image processing in fMRI.

This form of image registration can be performed with a variety of methods, two of them will be discussed in this paper. The goal is to find the values of the Affine transformation in order to best match a scan with a template. The first method is to represent image registration as a 3-dimensional Affine transformation of an image. Transformation values can often be found by using regression methods, other methods can be utilized that optimize different parameters between the scan and template, with the results varying in computational efficiency and accuracy. Although potentially inaccurate but simple, is least squares regression to find an optimal Affine transformation for scans; the simplicity of the method alone still proves worthwhile to explore and discuss. The second method, a more practical and commonly used approach is the use of Thin Plate Spline Radial basis functions. The process is more complex but the results are much better.

### 1.2 Background

By developing a matrix operator for image processors, the theoretical induced correlation can be quantified. Although the same goal could be accomplished through simulation, using a matrix operator has greater computational efficiency and accuracy.

When fMRI complex-valued data is collected in the spatial frequency domain, *k*-space, it is reconstructed into a recognizable image using a discrete inverse Fourier transformation. The reconstruction process can be represented in matrix form as, $v = \Omega f$ where, $\Omega$ is a single matrix operator, $f$ is a vector of observed values and $v$ is the reconstructed complex valued data [8]. If the original vector, $f$, had a mean , $\delta$, and a covariance, $\Gamma$, then the reconstructed image will have a mean and covariance of,

$$\mu = E[v] = \Omega\delta \text{ and } \Sigma = \text{cov}(v) = \Omega\Gamma\Omega^T$$

More often than not, it is desirable to conduct some processing on the reconstructed images to enhance analysis. It has been found that some common image processing techniques such as smoothing or sharpening induces a correlation. As before with reconstruction, image processing can be represented as a single matrix operator. If a matrix, $O$, is defined to be the collection of matrix operators necessary to perform reconstruction and processing then the same matrix representation can be used, $v = Of$ [7]. In this case the reconstructed image vector will have a mean, covariance and correlation between voxels of,

$$\mu = E[v] = O\delta,\ \Sigma = \text{cov}(v) = O\Gamma O^T \text{ and } \rho = corr(v) = DO\Gamma O^T D$$

where $D$ is a square matrix with diagonal elements formed by the reciprocal square root of the diagonal elements of $\text{cov}(v)$.

### 2.1 Method

Image registration techniques can be assigned into two categories: Rigid Body registration and Non-Rigid Body registration. Image processes, such as Motion Correction, are considered Rigid Body registration [1]. While Spatial Normalization qualifies as a Non-Rigid Body registration because surface deformations need to be accounted for. Since the scans are being matched to a template that may have features unique to the

scan the Non-Rigid Body registration is a far more complex problem. The process of Spatial Normalization can be viewed as the matching of two surfaces.

The purpose of this project is to create matrix operators for commonly used methods of Spatial Normalization and quantify the induced correlation if there is any. Two methods which are commonly used are Affine transformations given 12 parameters and Thin Plate Splines belonging to the set of Radial Basis functions.

### 2.2 Affine Transformation

Affine transformations are straight forward transformations to implement and derive. For the Process of Spatial Normalization a 3-dimensional Affine transformation is needed [2]. This 3-dimensional transformation can be represented as the combination of 12 parameters in a 4×4 matrix: 3 translations, 3 scales, 3 rotations and 3 shears. The transformation changes each image's intensity values and its respective coordinates using the aforementioned parameters.

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & S_x \\ t_{2,1} & t_{2,2} & t_{2,3} & S_y \\ t_{3,1} & t_{3,2} & t_{3,3} & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig.1. 3-dimensional Affine transformation matrix, $T$ . t's are scales, rotations and sheers. S' are translations.

The transformation matrix, $T$ , in its current form only clearly displays 3 of the 12 parameters, the translations. In order to view the other 9, the transformation matrix, $T$ , can be further broken down into the multiplication between three matrices and addition of one that display the scales, rotations and shears about the *x*, *y* and *z* axis. For the purposes of this paper, the *x* and *y* coordinates will represent the coordinates of an image's intensity values, *z*. The order in which the matrices are multiplied do matter, as rotations about the *x*, *y* then *z* axis is not necessarily the same as the rotations about the *z*, *x* and *y* axis.

$$T = R_x R_y R_z + S \text{ where,}$$

$$R_x = A * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_y = B * \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_z = C * \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S = \begin{bmatrix} 0 & 0 & 0 & S_x \\ 0 & 0 & 0 & S_y \\ 0 & 0 & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig.2. Transformation matrix, $T$ , as a function of translation matrix, $S$ , and rotation matrices; $R_x$ , $R_y$ $R_z$ rotated and scaled about their respective axes.

However, for the purposes of Spatial Normalization it is not necessary to distinguish what rotations and in what order they are being conducted since it is the values of the whole transformation matrix which is being optimized.

Once a transformation matrix is determined, the image transformation is represented as the multiplication of the transformation matrix with multiple vectors of image values and their respective coordinates,

$$Tf = f' \text{ where, } \quad f = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \end{bmatrix}^T \quad f' = \begin{bmatrix} x'_1 & y'_1 & z'_1 & 1 \end{bmatrix}^T$$

After the transformation is completed, integer valued coordinates may not be produced. Although this may not be an issue for the intensity values represented in the z coordinate, a Cartesian grid system is necessary for display and some form of interpolation will be necessary. The interpolation for new coordinate values will be discussed in Section 2.4.

With the structure established, the optimum values of the transformation matrix can be found using multiple methods. Most of which are dependent upon computational efficiency, computational memory, and stability of the problem.

Least squares regression seems like a logical solution to optimization of the values of transformation matrix, $T$, it is computationally efficient easy to implement but one can quickly run into the problem of poor conditioning. A Gauss-Newton iterative process would help with poor conditioning but is more computationally expensive and may end up getting caught in a local minimum rather than the global. However, no matter what process is used the implementation of this matrix operator remains the same.

The matrix operator for an Affine transformation will be similar to the implementation shown above. Although in the previous case only one transformation is being applied to one coordinate at a time. If the same transformation was to be applied to an entire image then it can be represented as such,

$$(I_n \otimes T) f = f' \text{ where, } \quad \otimes \text{ is the Kronecker Product}$$

$$f = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & x_2 & y_2 & z_2 & 1 & \cdots & x_n & y_n & z_n & 1 \end{bmatrix}^T$$

$$f' = \begin{bmatrix} x'_1 & y'_1 & z'_1 & 1 & x'_2 & y'_2 & z'_2 & 1 & \cdots & x'_n & y'_n & z'_n & 1 \end{bmatrix}^T$$

Allowing one transformation to be applied in one matrix multiplication. If local registration is conducted and different transformation matrices, $T_1, T_2 \cdots T_n$, are required then the transformation can be represented as such,

$$Tf = f' \text{ where, } \quad T = \begin{bmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & & \\ \vdots & & \ddots & \\ 0 & & & T_n \end{bmatrix}$$

### 2.3 Thin Plate Spline

A more appropriate method than Affine transformation is the use of Thin Plate Spline Radial basis functions. The Thin Plate Spline will minimize the amount of "bending energy" based on the distances between landmark points [5]. Unlike before, Thin Plate Splines perform better at matching surfaces that may not necessarily be similar. This is an important feature for this method in Spatial Normalization since

it is not uncommon if not always to find vastly different surfaces to match. A benefit of the radial basis functions is that they also are not just limited to just Thin Plate Splines and various other basis functions (Table 1) can be used for interpolation with very little change to the algorithm [6].

Table 1. Various Radial Basis Functions

| | |
|---|---|
| Linear | $\Phi(r) = r$ |
| Polynomial | $\Phi(r) = r^n$ |
| Thin Plate Spline | $\Phi(r) = r^n \log(r)$ |
| Gaussian | $\Phi(r, \varepsilon) = e^{-(\varepsilon r)^2}$ |

The 2-dimensional transformation by radial basis functions can be understood as the summation of the function of Euclidean distances multiplied by their respective weights and then added to the coordinates also multiplied by their respective weights,

$$x' = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i \Phi(\| (x_i, y_i) - (x, y) \|) \;\&\; y' = b_1 + b_x x + b_y y + \sum_{i=1}^{n} w_i \Phi(\| (x_i, y_i) - (x, y) \|)$$

This transformation can also be partitioned into a linear and non - linear parts and allows a matrix representation [4].

$$\begin{bmatrix} x \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a_1 \\ a_x \\ a_y \end{bmatrix} \quad K = \begin{bmatrix} \| (x_1, y_1) - (x, y) \| & \| (x_1, y_2) - (x, y) \| & \cdots & \| (x_1, y_n) - (x, y) \| \\ \| (x_2, y_1) - (x, y) \| & \| (x_2, y_2) - (x, y) \| & & \\ \vdots & & \ddots & \\ \| (x_n, y_1) - (x, y) \| & & & \| (x_n, y_n) - (x, y) \| \end{bmatrix}$$

& where,

$$\begin{bmatrix} y \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ b_1 \\ b_x \\ b_y \end{bmatrix} \quad P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}$$

Given the matrix representation of the problem, only the weights need to be solved for given landmarks and it is a much simpler problem than the one presented in the earlier transformation and can be solved as,

$$W_{x,a} = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} x \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } W_{y,b} = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} y \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Once the weights have been found, the coordinates can be transformed to the new ones. For just the transformation of one set of coordinates, this can be represented in matrix form,

$$\begin{bmatrix} x'_1 \\ y'_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_x & a_y & 0 & K_1 w^T \\ b_1 & b_x & b_y & 0 & K_1 w^T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$ where, $K_1$ is the first row of matrix $K$.

Note that it is the coordinates of image values changing while the image values remain constant. Once again, one may not obtain integer valued coordinates which are necessary to display an image. The interpolation necessary to convert to a Cartesian grid system will be discussed in section 2.4.

This matrix operator is very similar to Affine transformation operator with the exception of one extra constant. This is not a problem since the vector of coordinates and image values has multiple ones which can represent the addition of the extra constant. This implementation can also be expanded to the transformation of all coordinates as one matrix multiplication,

$$\begin{bmatrix} x'_1 \\ y'_1 \\ z_1 \\ 1 \\ x'_2 \\ y'_2 \\ z_1 \\ 1 \\ \vdots \\ x'_n \\ y'_n \\ z_n \\ 1 \end{bmatrix} = \begin{bmatrix} a_x & a_y & 0 & K_1 w^T & & & \cdots & & a_1 \\ b_x & b_y & 0 & K_1 w^T & & & & & b_1 \\ 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 1 & & & & & \\ & & & a_1 & a_x & a_y & 0 & K_1 w^T & \\ & & & b_1 & b_x & b_y & 0 & K_1 w^T & \\ & & & 0 & 0 & 0 & 1 & 0 & \\ \vdots & & & 0 & 0 & 0 & 0 & 1 & \\ & & & & & & & & \ddots \\ 0 & & & & & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \\ x_2 \\ y_2 \\ z_2 \\ 1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$$

As mentioned before, there are multiple ones for which the extra constant can be multiplied with, hence the wraparound of $a_1$ and $b_1$ in the first two rows.

### 2.4 Cartesian Grid Interpolation

Once new coordinates have been for the best fit between two images the coordinates may not be integer valued. Although the coordinates are Cartesian coordinates it is still necessary for us to obtain integers to correctly display 2-dimensional images. There are a multitude of image interpolation methods (Table 2) and many of them will work for the purposes of Spatial Normalization. Some of the interpolation methods are more involved but the majority of them try to develop new interpolate new values based on distance or some ratio of distance to the closest surrounding values.

Table 2. Various Interpolation Methods

| Bi Linear Interpolation |
|---|
| Median Interpolation |
| Barycentric Interpolation |
| Inverse Distance Weighted Interpolation |

For this paper, Inverse Distance Weighted interpolation was chosen for computational efficiency and simplicity while producing good results but the other methods do work, each with their own advantages and disadvantages. It is defined as,

$$z'_i = \frac{\sum_{t=1}^{n} \frac{1}{||(x_i, y_i) - (x_t, y_t)||^p} z_t}{\sum_{t=1}^{n} \frac{1}{||(x_i, y_i) - (x_t, y_t)||^p}} \text{ where,}$$

$p$ is the power

$(x_i, y_i)$ is the location to be estimated

$(x_i, y_i)$ is the location of known values

$z'_i$ is the interpolated image value

$z_p$ is the known image values

The summation of the distance between the interpolated value and the known points can be viewed as the weights, it can be seen that the weights decrease as the distance increases. This interpolation can then be rewritten as,

$$z'_i = \frac{h_1(x,y)z_1}{\sum_{p=1}^{n} h_p(x,y)} + \frac{h_2(x,y)z_2}{\sum_{p=1}^{n} h_p(x,y)} + \cdots + \frac{h_n(x,y)z_n}{\sum_{p=1}^{n} h_p(x,y)} \text{ where,}$$

$h_n(x,y)$ is the nth weight

This Inverse Distance Weighted Interpolation is based on the Shephard's method and not the modified Shephard's method. Though Shephard's method is susceptible to poor interpolation given large sets of points, this implementation will limit the interpolation to the four closest values.

Using the rewritten version of the interpolation the matrix representation is simple, the matrix operator will simultaneously interpolate the new z values and reassign the new coordinates.

$$\begin{bmatrix} 1 \\ 1 \\ z'_1 \\ 1 \\ 1 \\ 2 \\ z'_2 \\ 1 \\ \vdots \\ n \\ n \\ z'_n \\ 1 \end{bmatrix} = \begin{bmatrix} 1/x_1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1/y_1 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h_1(x,y)/\sum_{p=1}^{n} h_p(x,y) & 0 & 0 & 0 & h_2(x,y)/\sum_{p=1}^{n} h_p(x,y) & & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & & \\ & & & & 1/x_2 & & & & \\ & & & & & 2/y_2 & & & \\ & & & & & & h_1(x,y)/\sum_{p=1}^{n} h_p(x,y) & & \\ & & & & & & & & \\ \vdots & & & & & & & 1 & \\ & & & & & & & \ddots & \\ 0 & & & & & & & & h_1(x,y)/\sum_{p=1}^{n} h_p(x,y) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \\ x_2 \\ y_2 \\ z_2 \\ 1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$$

### 3.1 Results

Before beginning matrix multiplications for quantification of any potential correlation, a few details should be addressed; Matrix operator structure, dimensions and computation.

When observing these three matrix operators it is easy to notice that all three transformations are a unique solution to their respective problems. Affine transformation is the combination of the 12-parameters which best minimize the squared error between the template and image, Thin Plate Splines is a function of the distances between landmark points which best minimize the bending energy, and Inverse Distance Weighted interpolation is compiled of weights based on the distances between points. This means that for all three, the matrix operator will have different and unique values when matching different subjects to the same template. This may seem trivial, but this also leads to the fact that although the structure of the correlation between voxels may be predictable the degree of correlation is unpredictable unless each transformation from scan to template is analyzed individually.

Next, the dimensions of the matrix operators should be addressed. For simplicity, if we examine only the real parts of previously developed matrix operators, for instance the 2 – dimensional Fourier reconstruction operator mentioned earlier, they are always of size $n^2 \times n^2$ for any given image that is $n^2 \times n^2$. The three defined matrix operators in this paper are always of size $4n^2 \times 4n^2$ for any given image is $n^2 \times n^2$, due to the addition of a pair of coordinate points for each image value and a 1 to support the addition of any constants. This increase in matrix dimensions can make computation difficult, both time and storage may need to be taken into account.

However, the main purpose is to quantify the potentially induced correlation between voxels. The multiplication of these three matrix operators with their respective transposes will result in more information than we need; variance of coordinates, covariance between coordinates, covariance between coordinates and image values, etc. Leading to a correlation matrix of unnecessary information as well. If computational efficiency is valued it may be better to determine which rows and columns of the matrix operator are necessary to compute the covariance matrix beforehand and to multiply only when necessary. If not, after the multiplication of the operator and its transpose is complete the covariance matrix of image values will be embedded within and it is only an issue of extracting the correct values.

With this in mind, the potential covariance and correlation can be constructed. For the sake of organization, the three matrix operators will be defined as $M_A$, $M_T$ and $M_I$ for the Affine transformation, Thin Plate Spline and Inverse Distance Weighted interpolation, respectively (Table 4). While their covariance and correlation matrices will be defined as $Cov(A)\,\&\,Corr(A)$, $Cov(T)\,\&\,Corr(T)$ and $Cov(I)\,\&\,Corr(I)$ in the same respective order, assume that these covariance and correlation matrices have already been extracted from the larger covariance matrices mentioned before. The covariance and correlation can now be quantified (Table 3).

Table 3. Covariance & Correlation

| $Cov(A) = M_A M_A^T$ | $Cov(T) = M_T M_T^T$ | $Cov(I) = M_I M_I^T$ |
|---|---|---|
| $Corr(A) = D^{-\frac{1}{2}} Cov(A) D^{-\frac{1}{2}}$ | $Corr(T) = D^{-\frac{1}{2}} Cov(T) D^{-\frac{1}{2}}$ | $Corr(I) = D^{-\frac{1}{2}} Cov(I) D^{-\frac{1}{2}}$ |

Table 4. The matrix operators for Affine transformation, Thin Plate Spline and Inverse Distance Weighted interpolation given their original notation.

$$M_A = \begin{bmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & & \\ \vdots & & \ddots & \\ 0 & & & T_n \end{bmatrix}$$

$$M_T = \begin{bmatrix} a_x & a_y & 0 & K_1 w^T & & & \cdots & & a_1 \\ b_x & b_y & 0 & K_1 w^T & & & & & b_1 \\ 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 1 & & & & & \\ & & & a_1 & a_x & a_y & 0 & K_1 w^T & \\ & & & b_1 & b_x & b_y & 0 & K_1 w^T & \\ & & & 0 & 0 & 0 & 1 & 0 & \\ \vdots & & & 0 & 0 & 0 & 0 & 1 & \\ & & & & & & & & \ddots \\ 0 & & & & & & & & \end{bmatrix}$$

$$M_I = \begin{bmatrix} 1/x_1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1/y_1 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h_1(x,y)/\sum_{p=1}^{n} h_p(x,y) & 0 & 0 & 0 & h_2(x,y)/\sum_{p=1}^{n} h_p(x,y) & & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & & \\ & & & & 1/x_2 & & & & \\ & & & & & 2/y_2 & & & \\ & & & & & & h_1(x,y)/\sum_{p=1}^{n} h_p(x,y) & & \\ \vdots & & & & & & & 1 & \\ & & & & & & & & \ddots \\ 0 & & & & & & & & h_1(x,y)/\sum_{p=1}^{n} h_p(x,y) \end{bmatrix}$$

As mentioned before the structures are known but not the degree of correlation, the best way to demonstrate the correlation structure is through an illustrative example. This example is displayed in Figs. 3(a)-(c), with a 96×96 scan with obvious distortions and a 96×96 template.

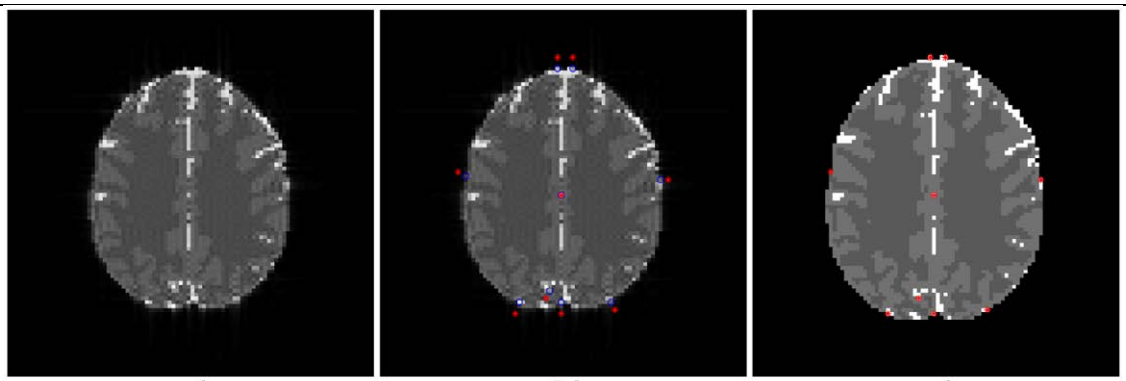
The images are matched with the template through either Affine transformation or Thin Plate Splines along with Inverse Distance Weighting interpolation if necessary. The matrix operator which transformed the image to the template is unique to this example, although the structure of the operator remains the same. Now that the matrix operators have been established, the covariance and correlation matrices can be produced as given Table 3. The correlation matrices are show in Fig. 4(a)-(c). Once again, the covariance and correlation matrices produced here are already of the reduced and desired size. Once correlation matrices have been determined the correlation between each image value and all other values

can be observed in Figs. 5(a)-(c). For this example the center image values correlation with all others have been superimposed on the image in Fig. 3(a), are displayed in Fig. 6(a)-(c).

It can be seen that the only operator that induces any correlation is Inverse Distance Weighted Interpolation. If we look back at how these matrix operators were formed it is not surprising that the Affine transformation and Thin Plate Splines do not induce correlation; The transformations are optimizations to fit one image to another and do conduct the transformation by weighting its surrounding areas.

The Inverse Distance Weighted interpolation is directly weighting its surrounding image values to build new ones. It becomes clear that correlation of no biological origin will be induced. This matrix operator would never be used on its own and would be paired with whatever registration method needs interpolation to a Cartesian grid system. What is most interesting about the correlation is that its structure is a reflection of where the weights have been placed. Since this is the case, the range of the induced correlation is actually determined by the range of the interpolation. In addition, the degree of correlation is then related to the power of the interpolation.

Figs. 3. (a) scan, (b) scan with landmarks &(c) template with landmark.




Figs. 4 (a) Correlation matrix from Affine Transformation (b) Correlation matrix from Thin Plate Spline Transformation (c) Correlation matrix from Inverse Distance Weighted Interpolation

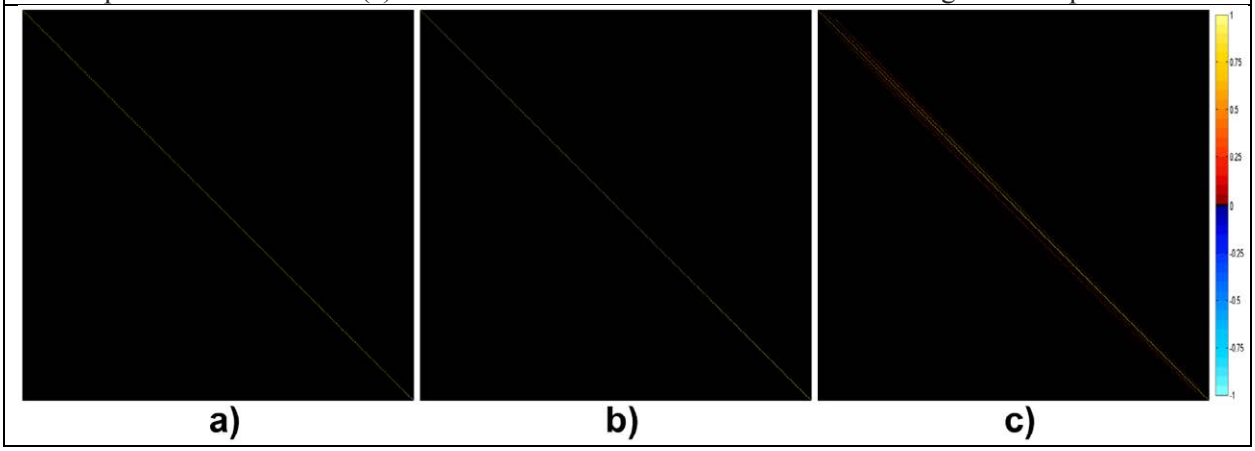


Fig. 5 (a) Correlation of center image value with others from Affine Transformation (b) Correlation of center image value with others from Thin Plate Spline Transformation (c) Correlation of center image value with others from Distance Weighted Interpolation

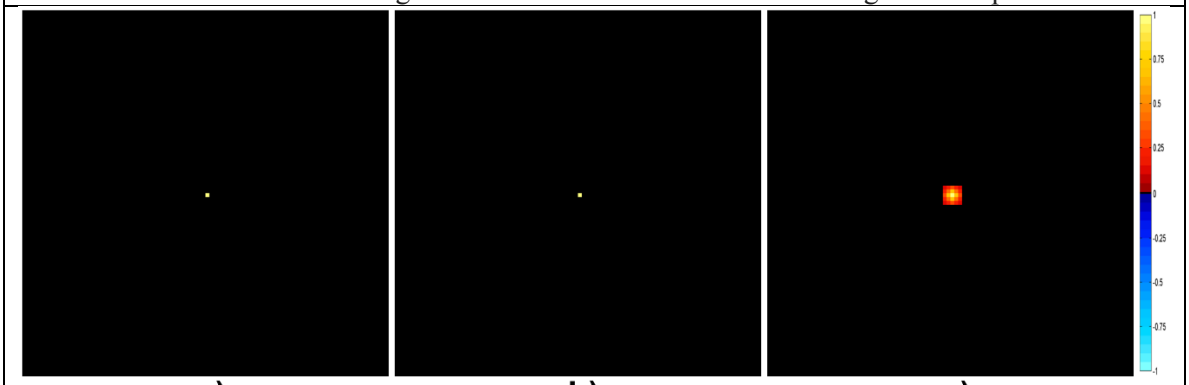
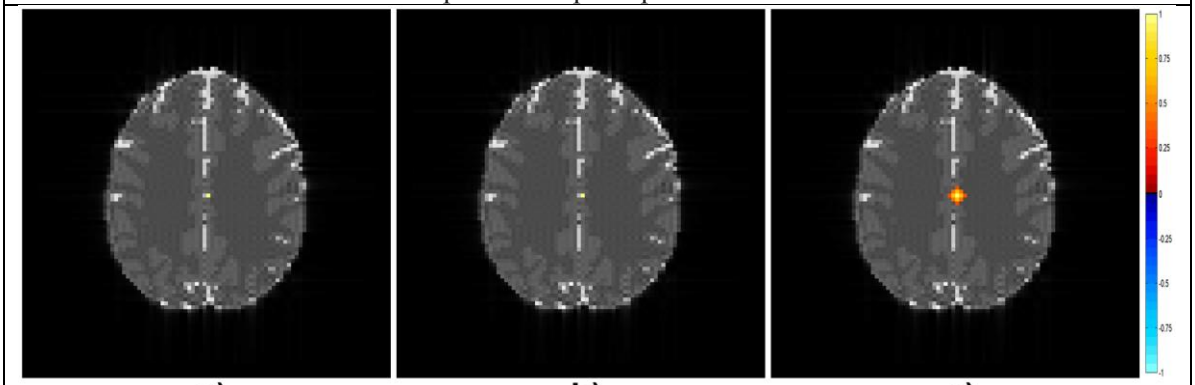



Fig. 6 (a) Correlation of center image value with other from Affine Transformation superimposed on scan (b) Correlation of center image value with other from Thin Plate Spline Transformation superimposed on scan (c) Correlation of center image value wither others from Distance Weighted Interpolation superimposed on scan




### 4.1 Conclusion

Spatial Normalization is not limited to just the methods described in this paper, each have their own advantages and disadvantages. However, these were chosen to best describe the overall process of Spatial Normalization.

The matrix operators for Affine transformation, Thin Plate Splines and Inverse Distance Weighted interpolation have all been defined. These are three commonly used image processors for the registration of an image to a template. Varying brain sizes and abnormalities makes the process of Spatial Normalization and any kind of registration a very useful tool. When images across multiple subjects are all registered to a template, it would be good understand if there is any correlation between image values.

It is observed that any correlation induced is due to interpolation of new values rather than the transformation to match two images together. There are many kinds of interpolation methods and this was chose for its simplicity. Looking more closely at different interpolation methods will give a much better understanding of potentially induced correlation

## Acknowledgements

This work was supported by National Institutes of Health research grant R21NS087450.

## References

[1] Ashburner J, Friston K. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*. 1999;7(4):254-266. doi:10.1002/(sici)1097-0193(1999)7:4<254::aid-hbm4>3.0.co;2-g.

[2] Ashburner J, Friston K. Rigid Body Registration. *Statistical Parametric Mapping*. 2007:49-62. doi:10.1016/b978-012372560-8/50004-8.

[3] Bookstein F. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1989;11(6):567-585. doi:10.1109/34.24792.

[4] Buhmann M. *Radial Basis Functions*. Cambridge: Cambridge University Press; 2003.

[5] Friston K, Ashburner J, Frith C, Poline J, Heather J, Frackowiak R. Spatial registration and normalization of images. *Human Brain Mapping*. 1995;3(3):165-189. doi:10.1002/hbm.460030303.

[6] Gianluca Donato and Serge Belongie. 2002. Approximate Thin Plate Spline Mappings. *Proceedings of the 7th European Conference on Computer Vision-Part III* (ECCV '02), Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen (Eds.). Springer-Verlag, London, UK, 21-31.

[7] Nencka A, Hahn A, Rowe D. A Mathematical Model for Understanding the STatistical effects of k-space (AMMUST-k) preprocessing on observed voxel measurements in fcMRI and fMRI. *Journal of Neuroscience Methods*. 2009;181(2):268-282. doi:10.1016/j.jneumeth.2009.05.007.

[8] Rowe D, Nencka A, Hoffmann R. Signal and noise of Fourier reconstructed fMRI data. *Journal of Neuroscience Methods*. 2007;159(2):361-369. doi:10.1016/j.jneumeth.2006.07.022.