

# Machine Learning Image Registration for Ultra-High Field MRI

Monday 15<sup>th</sup> May 2023

Oscar Windrath-Carr

Supervisors: Professor Chris Rodgers, Dr Krzysztof Klodowski

Wolfson Brain Imaging Centre, University of Cambridge

## Abstract

Image artefacts caused by subject motion are one of the dominant issues restricting image quality for ultra-high field magnetic resonance imaging (MRI). Motion correction of MRI head scans has been achieved to an extent through the FatNav approach. The FatNav approach takes multiple low-resolution images, called navigators, across the high-resolution host sequence. Host images are realigned retrospectively (after acquisition) using motion parameters between navigators. Motion parameters are extracted through classical image registration, a technique that takes two images of the same object and calculates the transform between them. All motion related artefacts cannot be avoided due to the correction taking place retrospectively. Rapid growth in machine learning over the past two decades invites the development of a learning-based image registration technique. Such a technique may provide both increased registration speed whilst achieving comparable or improved accuracy over the classical method. A rapid, robust, learning-based technique could allow the FatNav approach to be implemented prospectively (during acquisition), giving potential for improved motion correction. Synthetic datasets composed of FatNav-like head images were generated for use in both classical and learning-based registration methods. Classical registration was tested using the framework SimpleITK to give a reference point for speed and accuracy. Learning-based image registration was developed using a pre-built network from Project MONAI, a learning-based framework for various workflows in healthcare. Training and testing of this network was predominantly completed in 2D. The network was accurate in y translation and rotation parameters but not so in x translation. Computation was faster than classical registration by two orders of magnitude. Learning-based image registration in 3D was also developed and tested, showing encouraging results but with accuracy at this stage far from what is required. These results demonstrate the potential for developing a rapid, robust, learning-based 3D image registration method for implementation in the FatNav approach.

# Contents

<b>1 Introduction</b>	<b>3</b>
<b>2 Background</b>	<b>3</b>
2.1 Introduction to MRI	3
2.2 Image Artefacts	4
2.3 Approaches to Avoiding Motion Artefacts in MRI	4
2.4 Navigator Approach to Motion Correction	4
2.4.1 Navigators	4
2.4.2 FatNavs	4
2.4.3 Classical Registration	5
2.5 Machine Learning	6
2.5.1 Machine Learning Basics	6
2.5.2 Motion Correction with Machine Learning	7
2.5.3 Learning-Based Registration with Project MONAI	7
<b>3 Methods</b>	<b>7</b>
3.1 Overview	7
3.2 Generating Synthetic Data	8
3.3 Classical Registration	9
3.4 Machine Learning-Based Registration with Project MONAI	10
<b>4 Results</b>	<b>11</b>
4.1 Classical Registration	11
4.2 Machine Learning-Based Registration with Project MONAI	13
<b>5 Discussion</b>	<b>17</b>
5.1 Synthetic Data	17
5.2 Classical Registration	17
5.3 Machine Learning-Based Registration with Project MONAI	18
<b>6 Conclusions</b>	<b>19</b>
<b>References</b>	<b>20</b>
<b>Appendices</b>	<b>22</b>
A MRI Basics	22
B Parallel Imaging for the FatNav Approach to Motion Correction	24

# 1 Introduction

Image artefacts caused by subject motion are one of the greatest challenges facing MRI. These usually appear as ghost replicas of the image. Motion may also cause diffuse image noise. Artefacts and noise can result in reduced image quality and harm diagnostic and scientific relevance [1].

Motion artefacts and noise are especially severe in ultra-high field (UHF) MRI where there is greater potential for images of high resolution than with a clinical-grade scanner. UHF MRI has magnets with field strengths above 7T whereas clinical standard is below 3T.

Many attempts have been made to prevent, suppress, or correct for motion artefacts [2]. Reduction of image acquisition time leads to less development of artefacts, but compromises must be made on resolution, contrast, and signal-to-noise ratio (SNR) [1]. Other approaches exist but these are, at best, partial and there is room for improvement.

A successful approach for motion correction involves embedding navigators, quickly acquired, low resolution images during the host sequence, which can determine position of the patient's head at a certain point in the scan. The retrospective FatNav approach uses fat-selective excitations as navigators for correction of rigid motion after images are acquired [3]. Motion estimation between FatNavs is completed using image registration, a method to give motion parameters between two images of the head. These motion parameters then allow for retrospective correction of the host images.

The principal goal of the project is to develop a rapid and accurate registration technique based on machine learning to implement prospective (i.e. during image acquisition) FatNav-based motion correction of MRI head scans.

Section 2 provides background on basic MR theory, motion correction, and machine learning. The methods used in the project are outlined in Section 3. Results are presented in Section 4. Discussion is given in Section 5 with conclusions provided in Section 6.

## 2 Background

### 2.1 Introduction to MRI

Magnetic Resonance Imaging (MRI) has proven to be one of the most important diagnostic tools of medicine. Due to its versatility, sensitivity and wide range of tissue contrasts, MRI allows diseases throughout the body to be located and identified.

MRI relies on the phenomenon of nuclear magnetic resonance (NMR), induced and detected by radiofrequency coils and interactions of nuclear magnetization of non-zero spin nuclei with field gradients to image the body. Hydrogen nuclei are primarily targeted as they are abundant throughout the body principally in water and fat.

Modern MRI scanners rely on superconducting magnets, which cause a small fraction of protons in the body to align their magnetic moment with the field (on the order of ten excess aligned per million at 3T).

A broader classical description of MRI, including RF pulses, gradients, and imaging can be found in Appendix A.

## 2.2 Image Artefacts

As the resolution of MRI has improved, image quality is increasingly dominated by subject motion. Continuous motion can cause generalized blurring. Several small movements cause subtle ghosts which may be hard to spot [4].

## 2.3 Approaches to Avoiding Motion Artefacts in MRI

Various attempts have been made to avoid motion artefacts [1] [5]. These approaches can generally fall into one of three categories: motion prevention, artefact reduction and motion correction. Motion prevention can be attempted by patient immobilisation. Whilst this is widely used, it does not prevent some residual motion which is especially relevant for UHF MRI.

Artefact reduction can be accomplished through adjusting sequence parameters such as running a faster sequence [5]. However, this results in a lower resolution scan which does not fully utilise the capabilities of an UHF MRI scanner.

Motion correction is an approach which tracks patient motion during the scan and either adjusts acquisition parameters during the scan (prospective) or realigns data after the scan (retrospective).

## 2.4 Navigator Approach to Motion Correction

### 2.4.1 Navigators

A promising solution involves embedding ‘navigators’ – quickly acquired, low resolution images in the host sequence, which allow for locating the patient’s head at a certain point in the scan. Implementing retrospectively involves realigning the host images after acquisition guided by these navigator images. Prospective correction involves adjusting the acquisition of the host image during the scan. Retrospective correction is easier to implement, but it cannot always correct all motion related artefacts.

Extracting motion parameters from the navigators can be achieved in various ways. Each navigator can be compared with a reference navigator of the same type. Another option is to compare each navigator with a more complete reference signal [6]. Motion parameters can then be used for correction.

### 2.4.2 FatNavs

One choice of navigator is a ‘FatNav’ [3] [7]. The FatNav approach uses fat-selective excitations as navigators for retrospective correction of rigid motion.

Fat excitation leads to sparse images of the head (due to the signal primarily localized to sub-cutaneous fat and bone-marrow) which allows for high parallel imaging acceleration factors and hence a short scan time which can be accommodated in the dead time of many imaging sequences (Figure 1) [3].

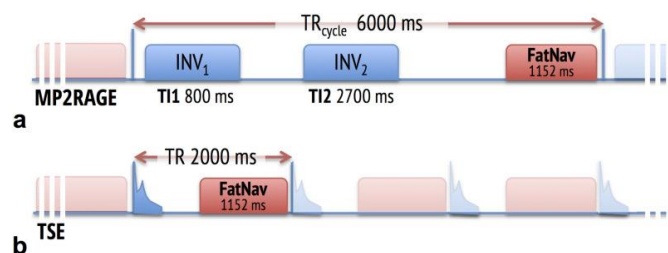


Figure 1: Pulse sequence diagrams of (a) the MP2RAGE sequence and (b) the TSE sequence, showing the relative timing of the 1152 ms 3D FatNav [3].

See Figure 2 for an example of a FatNav. See Appendix B for detail on parallel imaging for FatNavs.

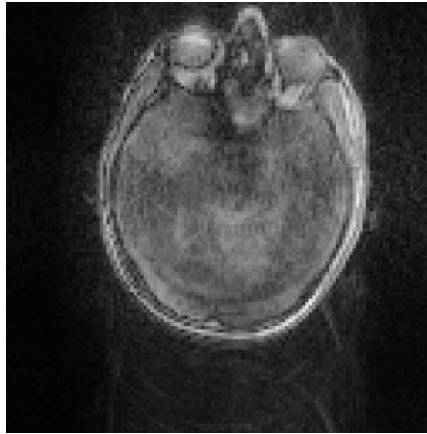


Figure 2: A slice from a real FatNav taken at WBIC (128 x 128 mm, 88 slices; ~1 s scan time). Notice noise and artefacts running top to bottom.

Figure 3 demonstrates the impact of the FatNav approach [3]. Whilst there is no independent measure of remaining bias in the motion correction estimates, the images visibly improved in sharpness. In principle, the FatNav approach could be implemented prospectively but this is restricted by limits on processor load and memory demands.

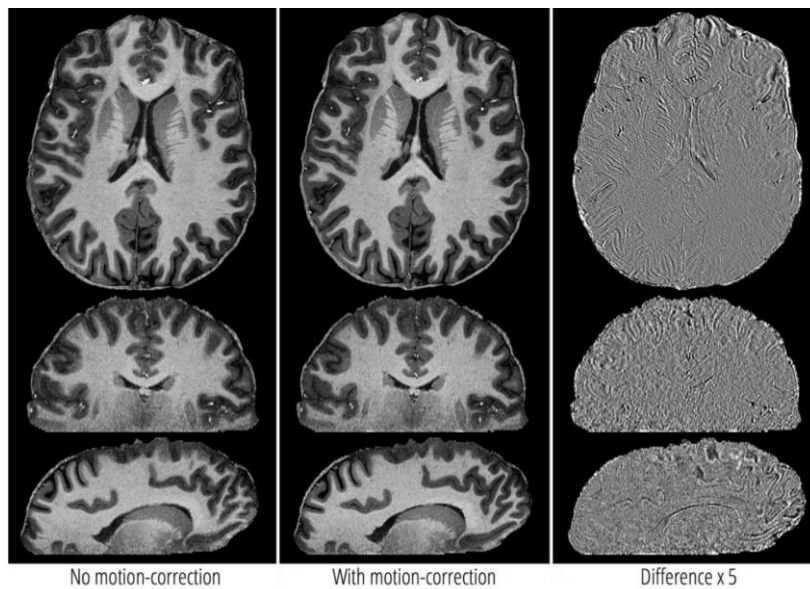


Figure 3: Retrospective motion correction results from the FatNav approach [3].

### 2.4.3 Classical Registration

Image registration is required to gauge the patient's head motion in the FatNav method. For two images of the head, registration gives motion parameters between them.

A common frame of reference must be set as the 'fixed image'. Other images in the set are called 'moving images'. The goal is to calculate the transform between each moving and fixed image.

Rigid registration is applied because there is negligible deformation between FatNav images of the head. Open-source frameworks are available for registration such as SimpleITK [8]. A registration method can be implemented by selecting several components: transformation, similarity metric,

optimizer, and interpolator. Using SimpleITK is straightforward and does not require a full understanding of how classical registration works.

## 2.5 Machine Learning

### 2.5.1 Machine Learning Basics

Machine learning is a field for developing methods to ‘learn’ based off data for a specific task without being explicitly programmed to do so.

Neural networks are a type of machine learning model inspired by biological neural systems. A neural network is a system composed of artificial neurons which make up layers of the network. A network is composed of an input layer, at least one hidden layer and an output layer (see Figure 4). These networks allow for non-linear data modelling determined by the connections between the neurons [9].

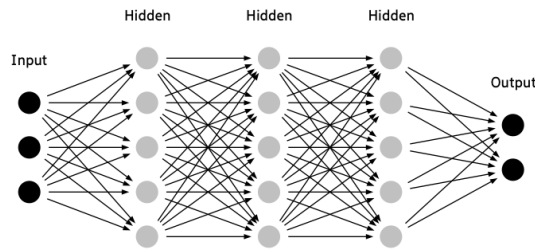


Figure 4: Example of a basic neural network with parameters and functions omitted [10].

A neuron’s output is determined by input, weights, and activation function. The input will be data, either the original input or the output of neurons from the previous layer. The weights determine the importance of each variable in the neuron’s function, a basic example is,

$$f(x_1, x_2, \dots, x_n) = w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (1)$$

where  $x_i$  are the variables,  $w_i$  are the weights and  $n$  is the number of inputs. Non-linear behaviour is accomplished through activation functions such as a sigmoid function,

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2)$$

output of the neuron’s function is passed through the activation function determining the output to a neuron in the next layer. For the sigmoid function, values are mapped between zero and one. This function gives a bias towards larger outputs. Other activation functions may be chosen depending on the task [9].

The network is trained for its purpose using a large dataset as alike the data it is being implemented for with as much variation as possible. Training can be supervised with labels for the data so the network output can be compared to the true values.

Accuracy is evaluated using a loss function such as mean squared error. The loss function is minimised over iterations of the dataset (called epochs) during training. For each neuron, the weights and parameters of the activation function are tuned. This is completed through gradient descent which determines the direction the network should take for loss to converge to a minimum [9].

When training is concluded, the network can be applied to unseen data. This allows for performance to be assessed.

### 2.5.2 Motion Correction with Machine Learning

Deep learning methods have recently been applied for motion correction in MRI. A neural network can be trained with a dataset containing motion-corrupted images as input and motion-clean images as labels [11]. Convolutional neural networks (CNNs) can detect outliers in motion-corrupted k-space and eliminate deformed k-space lines [11].

Acquiring real datasets for training requires long scan times, increasing the cost of such studies. Motion-simulation approaches are used for learning-based motion correction to generate synthetic training datasets [11].

### 2.5.3 Learning-Based Registration with Project MONAI

Project MONAI [12] is a PyTorch-based [13] framework for various workflows in healthcare including 2D and 3D registration.

Project MONAI provides various pre-built networks including GlobalNet, one suitable for affine image registration [14] [15]. Implementation of GlobalNet can be seen in the 2D unsupervised registration example [16]. For an image pair, the output of the network is a dense displacement field. A dense displacement field gives, for every position in the fixed image, the vector to move to match the moving image [17]. See Figure 5 for a 2D non-rigid example.

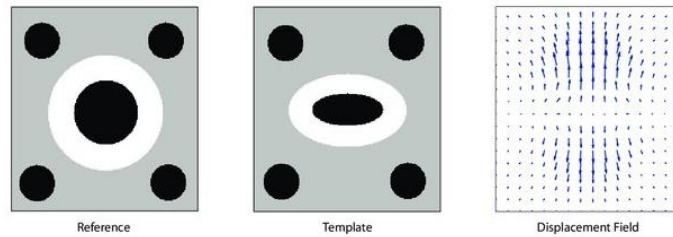


Figure 5: Dense displacement field for a non-rigid 2D transformation [17].

## 3 Methods

### 3.1 Overview

The main goal of the project was to develop a rapid and accurate image registration technique based on machine learning to be implemented in the FatNav approach to motion correction.

For consistency, the coordinate system shown in Figure 6 was used across the project. A skull mesh slice is inserted for reference.

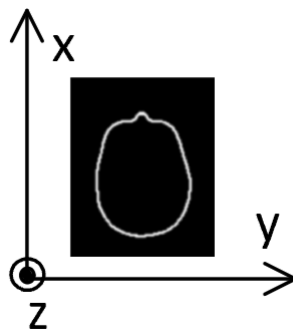


Figure 6: Axes convention for the project.

To assess performance of registration methods, a dataset was required. This dataset had to be made up of identical FatNav or FatNav-like 3D images which had been translated and rotated in space.

One image was augmented giving transformed images with ground truth motion parameters so that performance of a registration method could be assessed. Ground truth motion parameters also allowed for supervised training of a network on the dataset.

The next step was to implement a classical registration method. This was run on the dataset comparing results with ground truths to assess performance.

Machine learning-based registration was initially attempted through building a classification-type neural network using Keras [18]. Due to poor results and time constraints, this attempt was abandoned.

Learning-based registration was implemented using Project MONAI [12]. Building on 2D and 3D examples provided on the Project MONAI GitHub [16], a network was trained and evaluated for 2D datasets.

Project MONAI-based 2D registration was modified to work with 3D images. The network was trained and evaluated using a 3D dataset.

### 3.2 Generating Synthetic Data

The base image to generate synthetic data from was a skull mesh which is a near perfect 3D outline of a skull. This was chosen due to its similarity with the main feature of FatNavs being the sub-cutaneous fat of the head.

Image processing and augmentation used ImageJ [19]. The skull mesh was dilated and blurred to give a thicker, less sharp outline of the head which is more comparable to a FatNav image (Figure 7).

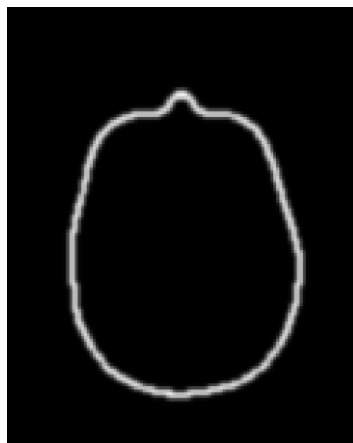


Figure 7: Skull mesh slice with dilation and blurring.

To augment the image, the skull mesh was translated and rotated (around the centre of the image) in conventional ZYX Euler angle order. Translation and rotation parameters were randomised within certain ranges. Ranges were defined from the Head Movement in UK Biobank study [20] (Figure 8). Translations were randomised within the range  $[-5, 5]$  mm and rotations were randomised within the range  $[-7, 7]$  degrees. For each augmented image, corresponding transformation parameters were saved.



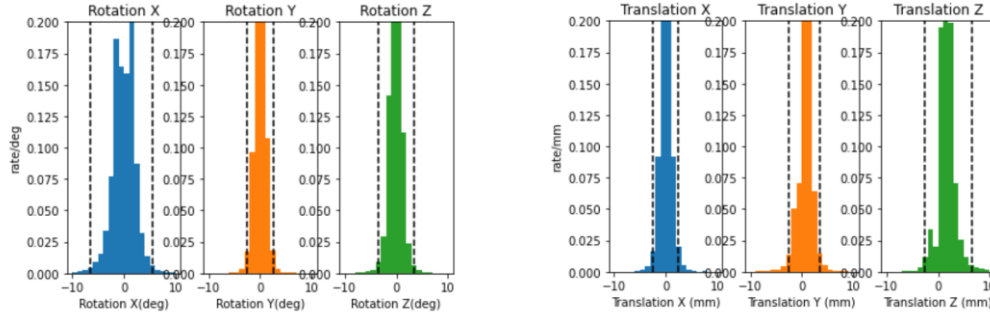


Figure 8: Range of motion in translations and rotations from a head movement study of 42,874 motion logs [20].

To generate a suitably realistic dataset, ‘GRAPPA’ – type artefacts were added, which are faint replicas running along the phase-encoding direction [21]. For FatNavs, they are found running top to bottom on the image. Due to the fixed direction, they were added to each image after augmentation. Two copies with brightness significantly reduced were added to each image. One copy was shifted up by half the height of the image, the other was shifted down by the same amount. See Figure 9 for an example of an augmented skull mesh with artefacts added.

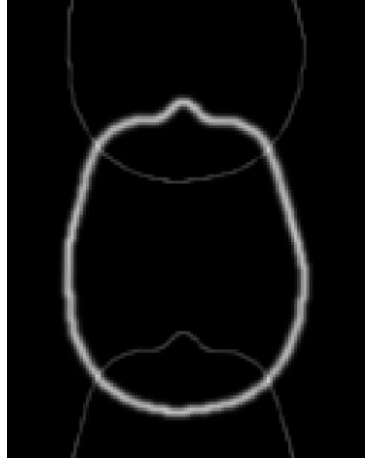


Figure 9: Augmented skull mesh slice with ‘GRAPPA’-type artefacts added.

The augmentation and artefact scripts were looped so that datasets of any size could be generated.

A similar script was written to generate synthetic 2D datasets. Datasets were generated from a skull mesh slice and real FatNav slices. ‘GRAPPA’-type artefacts and Gaussian noise were added to some skull mesh datasets.

33 synthetic FatNav datasets of 100 images each were created from 33 different FatNav slices. These FatNav slices were the middle slice of 33 FatNav images taken at the Wolfson Brain Imaging Centre (WBIC).

### 3.3 Classical Registration

Classical 3D registration was implemented in Python using SimpleITK [8]. The registration method used was from a SimpleITK notebook [22]. The method was mainly unchanged apart from removing the ‘multi-resolution framework’ and changing similarity metric to ‘MeanSquares’ because this resulted in more accurate registration of FatNav and skull mesh images.

The registration method was applied to 3D skull mesh data. Analysis was completed on results, plotting registration against ground truth motion parameters, and computing basic statistics (mean, median, mode and standard deviation) on normalised absolute differences for each motion parameter.

3D registration was run on real FatNav datasets from WBIC. This data consisted of 33 datasets containing between 5 and 11 images each, totalling 177 moving images with 33 fixed images.

The 3D registration method was then slightly modified to register 2D data. This allowed comparison with results from learning-based 2D registration. 2D registration was run on synthetic skull mesh artefact-free data and results were taken.

2D registration was run on the 33 real FatNav datasets where each 3D image was replaced by its middle slice. Registration was also run on 33 synthetic 2D FatNav datasets and results were taken.

### 3.4 Machine Learning-Based Registration with Project MONAI

Project MONAI-based implementation was written in Google Colaboratory (Colab) with a subscription to the premium service, Colab Pro+. Colab granted access to premium GPUs. Most of the time, an NVIDIA A100 40GB GPU was available which allowed for rapid training and testing of networks.

Image registration used GlobalNet, drawing from the 2D example to set network parameters and optimizer.

A loss function was required which used ground truth motion parameters to train the network. The result of the network had to be directly involved in the loss function to complete gradient descent. Using ground truth motion parameters, a dense displacement field was computed for each moving image. Mean squared loss was calculated between network output and dense displacement field for each training image.

Regularisation from Project MONAI was added to the loss and experimented with in relation to its magnitude.

To assess the performance of the network, a forward hook was used which gives an affine transformation matrix (Equation 3 where  $\theta$  is the anticlockwise rotation angle,  $T_x$  and  $T_y$  are the translations in  $x$  and  $y$  respectively) for each image in the test dataset. This matrix was used to attain the network's registered motion parameters for each image. Registered parameters were plotted against the ground truths. Statistics were also calculated as for the classical registration results.

$$\mathbf{T} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & T_x \\ \sin(\theta) & \cos(\theta) & T_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Training in all cases used a batch size of one so the network generalised better compared to larger batch sizes [23].

Firstly, the method was implemented for one translation and results were taken for a suitable dataset. It was then enhanced to accommodate two translations and then for translations and rotations with results being taken for both with suitable datasets.

Performance was assessed using 2D synthetic skull mesh data for varying training dataset size and number of epochs. For initial testing, datasets had no artefacts or noise added. A dataset with noise and 'GRAPPA'-type artefacts was also used. Then, the network was trained and tested using augmented FatNav data.

The test dataset was around 100 images for all cases.

Trained networks were run on the 33 real FatNav slice datasets from WBIC. Classical registration results for this data were used as ground truth values assuming classical registration gives reasonable results.

3D image registration was developed on top of the 2D method mostly just by adding a third dimension. Rotational order for forming dense displacement fields was ZYX (Equation 4 where  $T_i$  and  $\theta_i$  refer to translations and rotations in dimension  $i$  respectively). Training with 3D images was significantly more computationally demanding.

$$\mathbf{M} = TR_x R_y R_z = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

3D image registration was tested with results plotted.

## 4 Results

### 4.1 Classical Registration

Results for classical registration of the 3D synthetic skull mesh dataset are given in Figure 10 and Table 1.

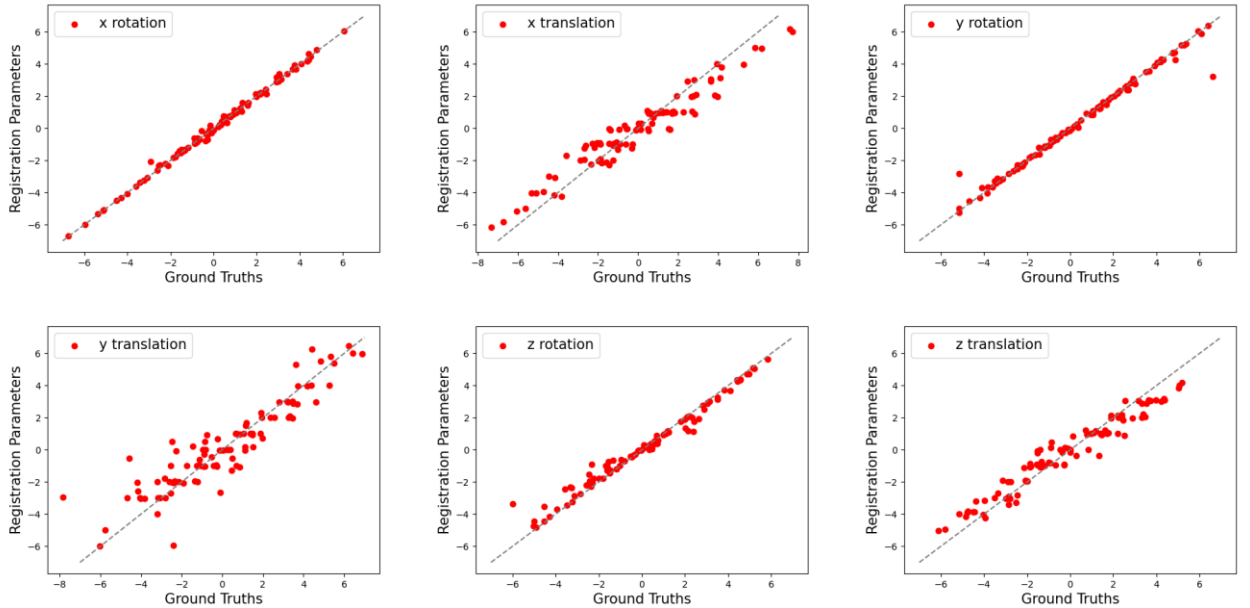


Figure 10: Registration parameters plotted against ground truths for each motion parameter from classical registration of 100 augmented 3D skull mesh images (rotations in degrees, translations in millimetres).

Table 1: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for classical registration of 100 3D augmented skull mesh images.

	x rotation	x translation	y rotation	y translation	z rotation	z translation
Mean	0.57	0.54	0.10	0.82	0.20	0.78
Median	0.04	0.29	0.03	0.34	0.10	0.27
Mode	0.00	0.00	0.00	0.01	0.00	0.01
SD	4.07	0.85	0.28	2.38	0.27	2.06

Results for classical registration of the 2D synthetic skull mesh dataset are given in Figure 11 and Table 2.

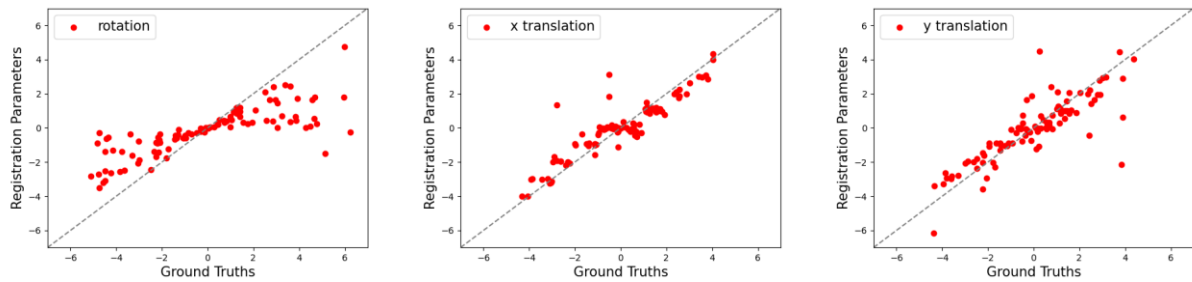


Figure 11: Registration parameters plotted against ground truths for each motion parameter from classical registration of 100 augmented 2D skull mesh images (rotations in degrees, translations in millimetres).

Table 2: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for classical registration of 100 2D augmented skull mesh images.

	Rotation	x translation	y translation
Mean	0.57	5.66	2.05
Median	0.54	0.35	0.37
Mode	0.01	0.01	0.01
SD	0.29	46.41	0.06

Results for classical registration of a synthetic 2D FatNav dataset are given in Figure 12 and Table 3.

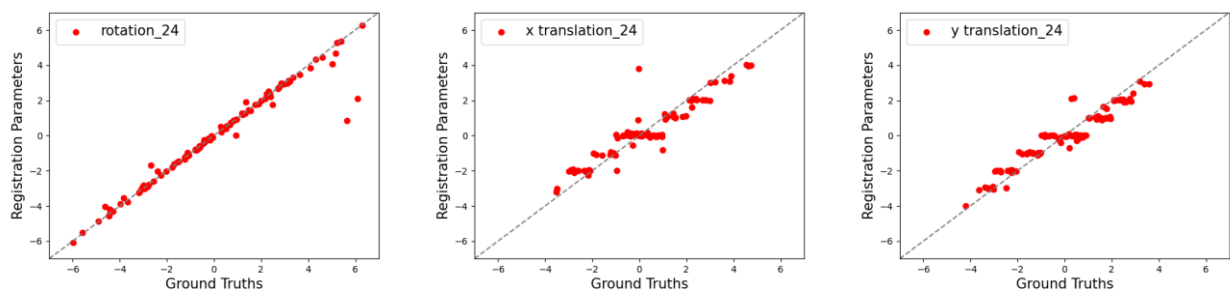


Figure 12: Registration parameters plotted against ground truths for each motion parameter from classical registration of 100 2D augmented FatNav images (rotations in degrees, translations in millimetres).

Table 3: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for classical registration of 100 2D augmented FatNav images.

	Rotation	x translation	y translation
Mean	0.11	2.73	0.63
Median	0.03	0.28	0.31
Mode	0.00	0.01	0.01
SD	0.20	20.17	0.99

Basic statistics on normalised differences between registration and ground truth motion parameters for all 33 datasets are given in Table 4.

Table 4: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for classical registration of 33 datasets, each containing 100 2D augmented FatNav images paired with one reference image.

	Rotation	x translation	y translation
Mean	8.68	4.35	3.61
Median	0.11	0.32	0.33
Mode	0.00	0.00	0.00
SD	320.00	75.61	37.92

For the real FatNav data, basic statistics on absolute values for each motion parameter are given for 3D and 2D registration in Tables 5 and 6 respectively.

Table 5: Basic statistics on absolute values of each motion parameter for 3D classical registration of 33 real FatNav datasets containing 5-11 images each (rotations in degrees, translations in millimetres).

	x rotation	x translation	y rotation	y translation	z rotation	z translation
Mean	0.54	0.71	0.88	3.73	1.90	1.10
Median	0.32	0.41	0.37	2.11	0.95	0.44
Mode	0.00	0.01	0.00	0.00	0.01	0.02
SD	0.74	1.26	1.39	15.75	2.19	1.50

Table 6: Basic statistics on absolute values of each motion parameter for 2D classical registration of 33 real FatNav datasets containing 5-11 images each (rotations in degrees, translations in millimetres).

	Rotation	x translation	y translation
Mean	0.97	1.47	2.53
Median	0.64	0.72	1.61
Mode	0.00	0.01	0.03
SD	1.02	4.31	5.02

## 4.2 Machine Learning-Based Registration with Project MONAI

For one translation in 2D, a network was trained on a dataset of 1000 skull mesh images for 30 epochs. Results for the test dataset are given in Figure 13.

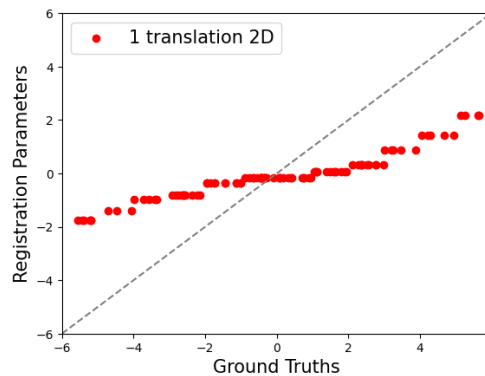


Figure 13: Registration parameters plotted against ground truths from learning-based registration of 100 2D augmented skull mesh images with translations in the y direction (in millimetres). The network was trained on a dataset of 1000 images for 30 epochs.

For translations in both directions in 2D, the network was trained on a dataset of 1000 images for 30 epochs. Results for the test dataset are plotted in Figure 14.

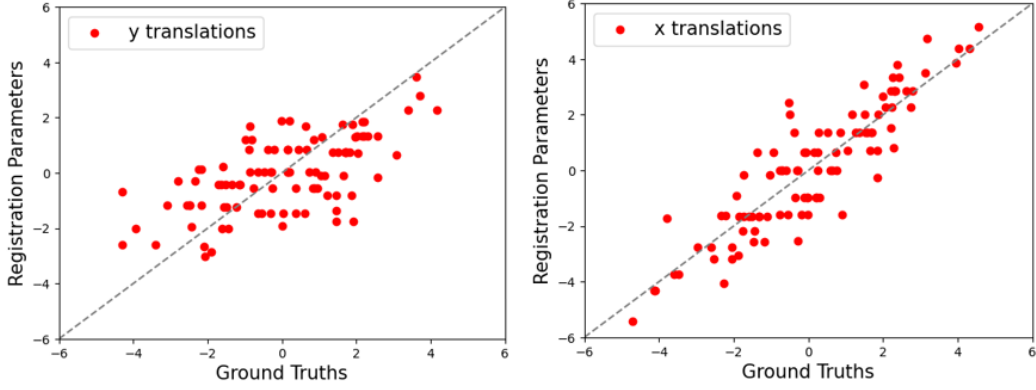


Figure 14: Registration parameters plotted against ground truths for each motion parameter from learning-based registration of 100 2D augmented skull mesh images with translations in the x and y directions (in millimetres). The network was trained on a dataset of 1000 images for 30 epochs.

For rotations in 2D, the network was trained on a dataset of 1000 images for 30 epochs. Results for the test dataset are plotted in Figure 15.

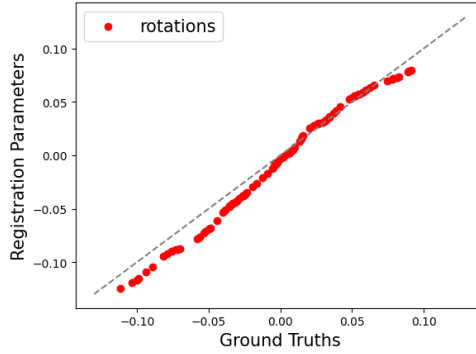


Figure 15: Registration parameters plotted against ground truths from learning-based registration on 100 2D augmented skull mesh images with rotations (in radians). The network was trained on a dataset of 1000 images for 30 epochs.

For rotations and translations in 2D, the network was trained on a dataset of 1000 images for 40 epochs. Results for the test dataset are given in Figure 16 and Table 7.

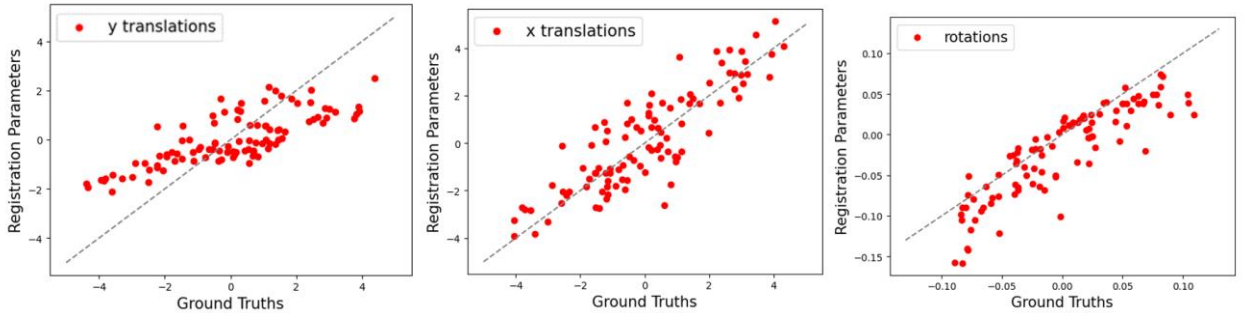


Figure 16: Registration parameters plotted against ground truths for each motion parameter from learning-based registration on 100 2D augmented skull mesh images with translations and rotations (rotations in radians, translations in millimetres). The network was trained on a dataset of 1000 images for 40 epochs.

Table 7: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for learning-based registration of 100 2D augmented skull mesh images with the network trained on a dataset of 1000 images for 40 epochs.

	Rotation	y translation	x translation
Mean	1.69	1.39	159.26
Median	0.45	0.70	0.49
Mode	0.01	0.01	0.01
SD	5.50	1.82	1571.49

The network was trained on the previous dataset for 60 epochs. Results for the test dataset are given in Figure 17 and Table 8.

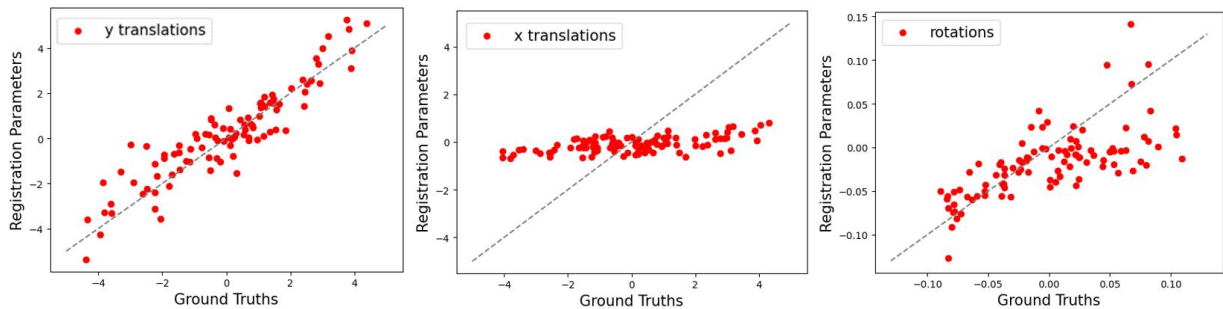


Figure 17: Registration parameters plotted against ground truths for each motion parameter from learning-based registration on 100 2D augmented skull mesh images (rotations in radians, translations in millimetres). The network was trained on a dataset of 1000 images for 60 epochs.

Table 8: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for learning-based registration of 100 2D augmented skull mesh images with the network trained on a dataset of 1000 images for 60 epochs.

	Rotation	y translation	x translation
Mean	2.47	1.05	38.90
Median	0.64	0.48	0.92
Mode	0.00	0.00	0.03
SD	9.56	2.15	376.33

The results above used skull mesh data without artefacts or noise added.

The network was trained on a 1000 image skull mesh dataset with translations, rotations, ‘GRAPPA’-type artefacts, and noise for 40 epochs. Results for a test dataset are given in Figure 18 and Table 9.

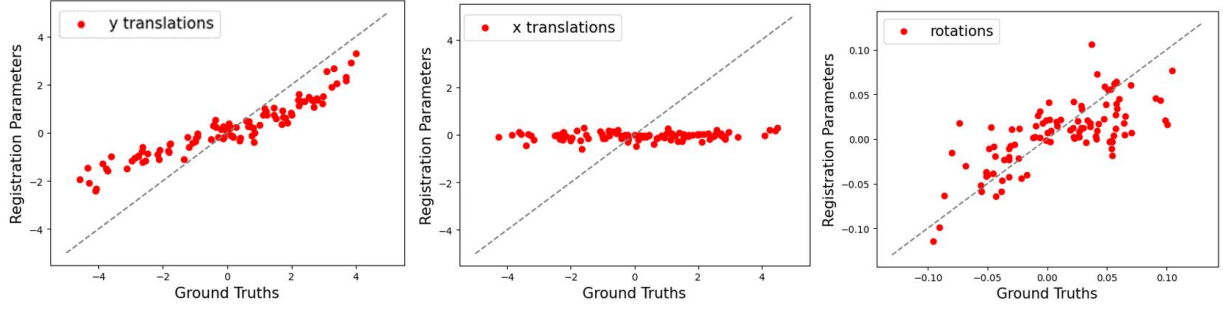


Figure 18: Registration parameters plotted against ground truths for each motion parameter from learning-based registration on 100 2D augmented skull mesh images with ‘GRAPPA’-type artefacts and noise (rotations in radians, translations in millimetres). The network was trained on a dataset of 1000 images for 40 epochs.

Table 9: Basic statistics on the normalised absolute differences for each parameter where normalisation is relative to the ground truth parameter. This is for learning-based registration of 100 2D augmented skull mesh images with ‘GRAPPA’-type artefacts and noise. The network was trained on a dataset of 1000 images for 40 epochs.

	Rotation	y translation	x translation
Mean	1.43	1.22	1.19
Median	0.59	0.61	1.0
Mode	0.01	0.08	0.09
SD	3.62	3.99	1.26

For augmented FatNav data, the network was trained on 30 images from each of the 33 datasets for 40 epochs. The test dataset was composed of 3 unseen images from each dataset, results are given in Figure 19.

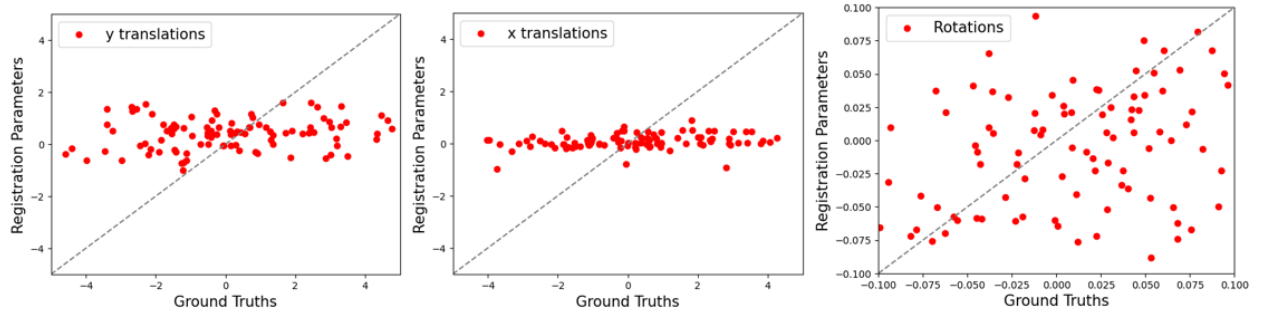


Figure 19: Registration parameters plotted against ground truths for each motion parameter from learning-based registration on 99 2D augmented FatNav images (rotations in radians, translations in millimetres). The network was trained on a dataset of 990 images for 40 epochs.

For the 3D method, the network was trained on a dataset of 300 augmented images, with ‘GRAPPA’-type artefacts, for 100 epochs. Results for a test dataset are plotted in Figure 20.



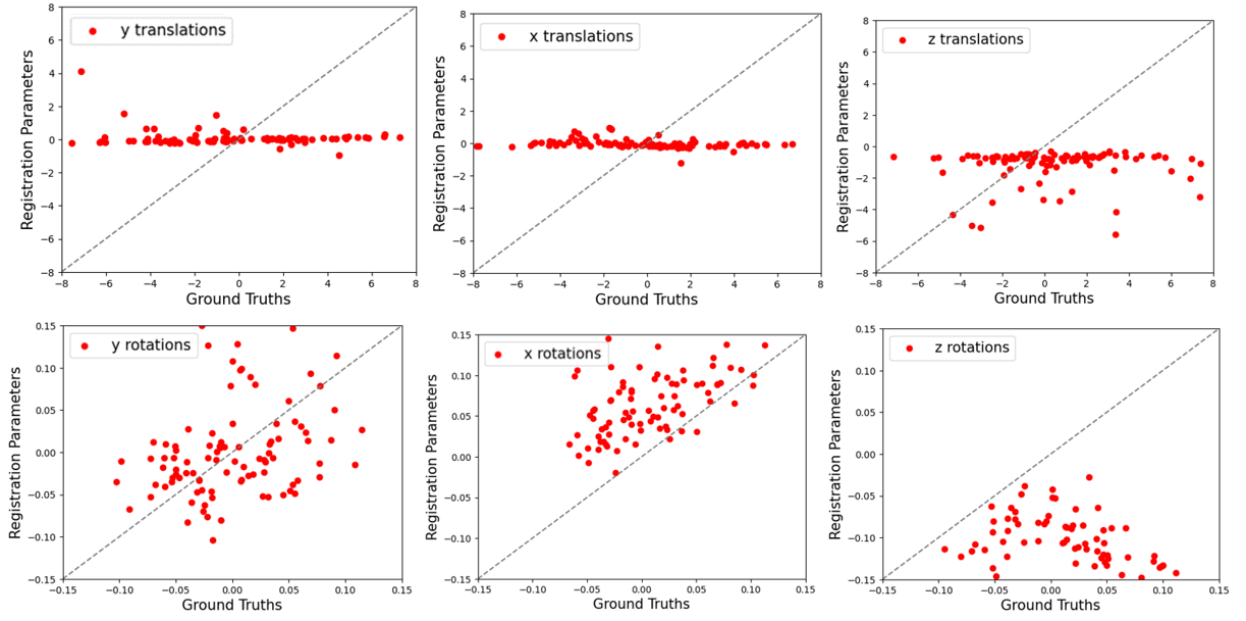


Figure 20: Registration parameters plotted against ground truths for each motion parameter from learning-based registration on 100 3D augmented skull mesh images with ‘GRAPPA’-type artefacts (rotations in radians, translations in millimetres). The network was trained on a dataset of 300 images for 100 epochs.

## 5 Discussion

### 5.1 Synthetic Data

The aim with the synthetic data was for the images to have the most defining features of FatNav images.

With the outline of the skull and ‘GRAPPA’-type artefacts as discussed in Section 3.2, the data was fairly comparable to real FatNav images. The main downfall was the absence of grey matter inside the head (compare Figure 2 with Figure 9). Due to time constraints, adding this feature was not attempted.

2D datasets generated from augmenting real FatNav slices provided variation to train a learning-based registration network.

### 5.2 Classical Registration

3D classical registration was run in PyCharm on a Microsoft Surface Pro (5<sup>th</sup> gen) laptop. This took  $20 \pm 2$  minutes to run for 100 images.

3D registration of skull mesh images gave strong positive correlation between registered and ground truth motion parameters. Rotation and translation in y are the most and least accurate respectively. Visibly in the graphs (Figure 10), rotations had more accurate registration than translations.

Whilst accuracy of registration may be adequate for motion correction, the time taken of  $12 \pm 1$  seconds per moving image is not suitable for prospective correction.

2D classical registration was run in PyCharm on a Microsoft Surface Pro (5<sup>th</sup> gen) laptop. Results (Figure 11) showed less accuracy than for 3D classical registration.

2D registration was also run in Google Colab to provide one-to-one comparison to learning-based methods using the same hardware. This took 11.2 seconds for 100 augmented skull mesh images.

Results for 2D registration of augmented FatNav slices (Figure 12, Tables 3 and 4) had more anomalies in values and variation in performance between datasets compared to skull mesh registration. This was anticipated with FatNav data containing artefacts, noise, and grey matter. There are also likely to be images where the head is cropped due to a large transformation. These factors led to large values for mean and standard deviation seen in Table 4.

As mentioned in Section 5.1, augmenting real FatNav data leads to the orientation of artefacts and noise not being preserved in the phase-encoding direction. This may have affected registration performance for the augmented FatNav datasets, but it is not possible to quantify this.

### 5.3 Machine Learning-Based Registration with Project MONAI

The aim with Project MONAI was to train a neural network to complete image registration with comparable or improved accuracy and to be faster than classical registration.

Results from translations in one direction have been classified into certain values giving the graph in Figure 13. This behaviour did not carry over into other trained networks.

Results for two translations in Figure 14 show promise although accuracy is not as good as for classical registration.

Training the network for just rotations gave good results (Figure 15). The network seems more suited to registering just rotations than translations.

For translations and rotations in 2D, the network was trained with various number of epochs, dataset size and dataset type. Beyond certain values, larger datasets and higher number of epochs led to worse results. For the same number of epochs (e.g. 50), a training dataset of 1000 images led to more accurate results on test data than 10,000. For a training dataset of 1000 images, 50 epochs of training gave more accurate predictions than 100. These findings are likely the network experiencing overfitting, when the network fits too well to training data so it cannot generalise well to unseen data [24]. For too many epochs, the network will learn too well each training image and motion parameters. Overfitting from larger training datasets is likely due to the dataset having images that are identical within an approximation (e.g. within  $0.5^\circ$  of rotation and 0.5mm of translations). The network will see approximately the same image multiple times within an epoch of training resulting in overfitting.

Trial and error led to a training dataset of 1000 images and number of epochs in the range 30-60 being used. Figures 16 and 17 demonstrate variation in accuracy with number of epochs. From these figures and Tables 7 and 8, comparable accuracy with classical registration is seen for rotation and y translation parameters (Figure 11 and Table 2). Although the x translation plot has positive correlation in Figure 16, Table 7 shows this is misleading because there is large standard deviation and mean in the absolute normalised difference. Further investigation is required to uncover the cause of this discrepancy between x and y translation registration accuracy.

Trained networks in Figures 16 and 17 took 0.25 and 0.23 seconds respectively to complete registration on the test dataset. This is approximately 2% of the time taken for classical registration of the same dataset run on the same hardware. Learning-based registration is faster than classical registration by two orders of magnitude in the 2D case.

2D registration on skull mesh data with ‘GRAPPA’-type artefacts and noise gave similar results to those with clean images for rotations and y translations. Although the statistics in Table 9 suggest that the x

translation parameter has been registered more accurately than previous runs, Figure 18 shows that this is not the case. Addition of artefacts and noise did not have a large effect on these results.

Visible from results on augmented FatNav data (Figure 19), the network struggled to learn to register FatNav slices. This is likely due to reduced SNR and addition of specific noise, artefacts and grey matter over the skull mesh slices. There may also be cropping as mentioned in Section 5.2.

Trained networks were run on the real 2D FatNav datasets and results compared with classical registration values as ground truths. These tests did not lead to any promising results. This was anticipated due to poor results on augmented FatNav data.

3D classical registration on the real FatNav data (Table 5) shows significant through plane movement (in z). Within a 2D dataset, the head may effectively undergo deformation between images. This would have influenced results of the trained networks. Additionally, the effect of this movement on 2D classical registration is visible in the discrepancy for x and y translation values between Tables 5 and 6.

Developing 3D image registration using Project MONAI was restricted by time and computational power. Most of the allocated time was spent developing learning-based 2D registration. The 3D image registration method was not able to handle many more than 300 training images due to limitations on GPU RAM. Results (Figure 20) show no correlation between registration and ground truth parameters for translations, but rotation results have slight positive correlation, particularly for y. 3D registration using Project MONAI requires further investigation with larger training datasets paired with greater computing resources.

## 6 Conclusions

The goal of the project was to develop an accurate and rapid learning-based image registration technique for FatNavs. This required a registration algorithm which could give motion parameters between two FatNav images through a suitably trained network.

Classical methods for image registration were used as a basis to learn from and compare with when developing learning-based methods. Synthetic data allowed for performance of classical methods to be assessed due to the motion parameters being known.

Learning-based image registration was developed through Project MONAI [12], the most suited open-source framework found. Results with 2D synthetic data after training showed comparable accuracy to classical registration in y translation and rotation parameters, with registration faster by two orders of magnitude. Accuracy was not comparable in the x translation parameter.

A learning-based image registration method again based on Project MONAI was developed for 3D images. This gave encouraging results, but accuracy was far from what is required for the FatNav approach to motion correction.

The future of this work will involve further development of 3D learning-based registration using larger datasets and greater computational power. The method must be validated on real unseen FatNav data before being implemented prospectively through the FatNav approach in a 7T scanner. This could lead to significant improvement in image quality and diagnostic and scientific relevance of images.

## Acknowledgements

I would like to express my appreciation to the supervisors of this project, Professor Rodgers and Dr Klodowski, for their guidance, support and invaluable suggestions.

## References

- [1] Godenschwegerv F. et al., 2016, “Motion correction in MRI of the brain”. *Phys Med Bio.*, 61(5): R32-R56.
- [2] Julian M. et al., 2013, “Prospective motion correction in brain imaging: A review”. *Magn Reason Med.*, 69(3): 621-36.
- [3] Gallichan D., Marques J.P., Gruetter R., 2016, “Retrospective Correction of Involuntary Microscopic Head Movement Using Highly Accelerated Fat Image Navigators (3D FatNavs) at 7T”. *Magnetic Resonance in Medicine*, 75: 1030-1039.
- [4] McRobbie D.W., Moore E.A., Graves M.J., Prince M.R., 2007, “MRI From Picture to Proton Second Edition”. Cambridge University Press, Cambridge.
- [5] Zaitsev M., Maclaren J., Herbst M., 2015, “Motion Artifacts in MRI: A Complex Problem With Many Partial Solutions”. *JMRI*, 42: 887-901.
- [6] Gretschi F., 2020, “Fat navigators based retrospective motion correction strategies for brain magnetic resonance imaging”. EPFL.
- [7] Klodowski K., Halabrin J., Dragonu I., Rodgers C., 2022, “Prospective 3D FatNav motion correction for 7T Terra”. *Proc. ISMRM2022*, #0860.
- [8] Beare R., Lowekamp B. C., Yaniv Z., 2018, “Image Segmentation, Registration and Characterization in R with SimpleITK”. *J Stat Software*, 86(8).
- [9] IBM. “What is a neural network?”. Available at: <https://www.ibm.com/topics/neural-networks> (Accessed 12 May 2023).
- [10] Ganesh S. K., 2021. “Awesome Drawing tools for Neural Net Architecture”. Available at: <https://www.kaggle.com/getting-started/253300> (Accessed 12 May 2023).
- [11] Lee S., Jung S., Jung K.J., Kim D.H., 2020 “Deep Learning in MR Motion Correction: a Brief Review and a New Motion Simulation Tool (view2Dmotion)”. *iMRI* 24: 196-206.
- [12] Jorge Cardoso M. et al., 2022, “MONAI: An open-source framework for deep learning in healthcare”. *arXiv:221.02701*.
- [13] Paszke A. et al., 2019, “PyTorch: An Imperative Style, High-Performance Deep-learning library”. *arXiv:1912.01703*.
- [14] Thorsten F. et al., 2019, “U-Net: deep learning for cell counting, detection and morphometry”. *Nature Methods* 16, 67-70.
- [15] Yipeng H. et al., 2017, “Label-driven weakly-supervised learning for multimodal deformable image registration”. *arXiv:1711.01666*.
- [16] Project MONAI GitHub, 30 January 2023. “2D Xray registration demo”. Available at: [https://github.com/Project-MONAI/tutorials/blob/main/2d\\_registration/registration\\_mednist.ipynb](https://github.com/Project-MONAI/tutorials/blob/main/2d_registration/registration_mednist.ipynb) (Accessed 12 May 2023).

- [17] Schwarz L. A., 2007, “Non-rigid Registration Using Free-form Deformations”. TUM.
- [18] Ketkar N., 2017, “Introduction to keras”. Deep learning with python: a hands-on introduction: 97-111.
- [19] Abramoff M.D., Magalhães P.J., Ram S.J., 2004, “Image processing with ImageJ”. Biophotonics international 11(7): 36-42
- [20] Hess A.T. et al., 2022, “Head movement in UK Biobank, analysis of 42,874 fMRI motion logs”. University of Oxford. Available at: [users.fmrib.ox.ac.uk/~ahess/MoCo2022.pdf](https://users.fmrib.ox.ac.uk/~ahess/MoCo2022.pdf).
- [21] Wikipedia, 20 February 2023. “MRI artifact”. Available at: [https://en.wikipedia.org/wiki/MRI\\_artifact](https://en.wikipedia.org/wiki/MRI_artifact) (Accessed 12 May 2023).
- [22] Yaniv Z., Lowekamp B.C., Johnson H.J., Beare R., 2018, “SimpleITK Image-Analysis Notebooks: a Collaborative Environment for Education and Reproducible Research”. J Digit Imaging, 31(3): 290-303.
- [23] Shen K., 2018. “Effect of batch size on training dynamics”. Available at: <https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e> (Accessed 12 May 2023).
- [24] Dietterich T., 1995, “Overfitting and undercomputing in machine learning”. ACM computing surveys, 27(3): 326-327
- [25] Chappell M., 2020, “Short introduction to MRI Physics for Neuroimaging”. Independently published.
- [26] Elster A.D., 2021, “Questions and Answers in MRI”. Available at: <https://mriquestions.com/index.html> (Accessed 19 November 2022).

## Appendices

### A MRI Basics

Modern MRI scanners rely on superconducting magnets causing a small fraction of protons in the body to align their magnetic moment with the field.

A radiofrequency (RF) pulse generated by transmitter coils, shifts proton net magnetization out of alignment with the field (Figure 21) by a given angle dependent on the power of the RF pulse. This excitation process will only be successful if the radiofrequency matches the resonant, Larmor, frequency of the nuclei.

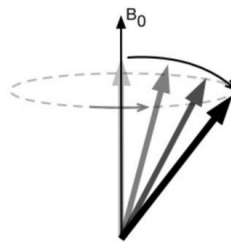


Figure 21: An RF pulse tips the magnetisation of a nuclei out of alignment with the field resulting in a rotating transverse component (given the radiofrequency matches the Larmor frequency) [25].

After the RF pulse, longitudinal magnetization (along the static field direction) recovery occurs in an exponential fashion with a time constant  $T_1$  (called  $T_1$  relaxation). Transverse magnetization decaying after an RF pulse is known as  $T_2$  relaxation and decays with time constant  $T_2$  [26]. After the RF pulse, the rotating transverse vector component produces an oscillating magnetic field which induces a small current in receiver coils (a free induction decay, FID). Due to inhomogeneities in practical static magnetic fields, the Larmor frequency varies across the body resulting in the time constant for  $T_2$  decay to be called  $T_2^*$  (always shorter than  $T_2$ ).  $T_1$  and  $T_2$  values vary widely between different tissues which is one factor contributing to the great soft tissue contrast of MRI.

### Gradients

Figure 22 outlines a section of a typical 2D spin echo sequence. A spin echo is the signal produced by two successive RF pulses. An initial  $90^\circ$ -pulse tips spins into the transverse plane resulting in a free induction decay. Shortly after, a  $180^\circ$ -pulse ‘flips’ the spins in the transverse plane. This flip essentially allows for regeneration of spin phase information producing a spin echo which can be seen in Figure 23 [26]. Note that different flip angles may be used to produce spin echoes, they do not have to be  $90^\circ$  and  $180^\circ$  [26].

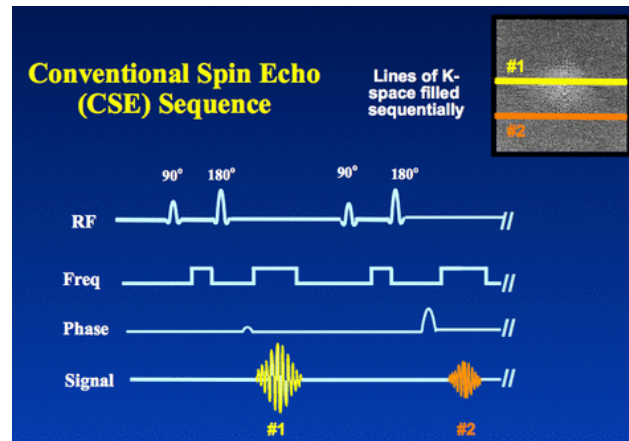


Figure 22: Conventional Spin Echo sequence [26].

Localisation of MR signals in the body to produce images is achieved by generating small, short-term spatial variations in magnetic field strength across the patient. These are referred to as imaging gradients and gradient fields are produced by gradients coils (one for each of three orthogonal spatial directions).

Due to precession frequency being dependent on the magnetic field strength, different spatial locations become associated with different precession frequencies. The regions where the precession frequencies match the RF pulse frequency will experience excitation. MR signal can be located as the Larmor frequencies across the field and various RF pulses applied are known (see Figure 22). Certain slices of the body can be imaged by applying the corresponding range of RF frequencies to match the Larmor frequency range in this slice (due to the variation induced by the applied gradient) [25]. The slice selective gradient is switched on during the first RF pulse for each line of k-space filled in Figure 22. Each k-space line is encoded with a different phase encoding gradient.

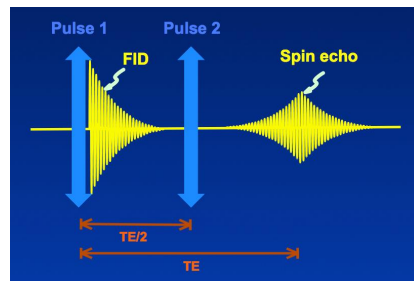


Figure 23: Creation of a spin echo by two RF pulses [26].

The signal received will be a mixture of all the different frequencies and phases. The space for this raw data is called k-space. Data can be acquired in any order such as line-by-line, spirally or radially. A Fourier Transform allows us to convert the k-space data into an image.

## Imaging

In general, images have contrast which depends on  $T_1$ ,  $T_2$  or the proton density (PD). Scans may be weighted to produce images depending on one of these three factors by setting the timing values called TR (repetition time) and TE (echo time).

MR images are acquired digitally. They are described as a matrix of pixels in 2D or voxels (volume elements) in 3D. The MR signal is an oscillating (analogue) electric voltage in the receiver coil which is converted to a digital signal before reconstructed into the final image. The signal is digitized by an ADC (analogue-to-digital converter).

Pixels in MR images are organised into matrices. The input matrix dimensions on a scanner sets final image size. As scanning takes place, k-space is filled through a sampling method. One dimension of the matrix determines how many times the sequence must be repeated thus defining the scan time. This is known as the phase-encoding (PE) axis of the matrix and the other dimension is known as the frequency-encoding axis (FE).

## **B Parallel Imaging for the FatNav Approach to Motion Correction**

Parallel imaging involves multiple coils with varying spatial location to assess the location of an MR signal using coil sensitivity information. Signals from individual coils are processed simultaneously along separate channels (without parallel imaging individual outputs are combined before being processed). This reduces the number of phase-encoding steps and hence speeds up imaging resulting in scan times for FatNavs being negligible relative to the host sequence [3].

Limitations of parallel imaging must be considered. Regarding use of parallel imaging for FatNavs, we only need good enough images to pin down the location of the patient, so reduced image quality is not so relevant. Furthermore, due to sparsity of fat in the head, degradation only becomes significant at higher acceleration factors (degradation of image quality increases with acceleration factor). Note that in this study [3], the host sequences are taken without parallel acceleration.