Technical Report: the 7th Brain(s) Contest by FUJIFILM AI Academy Brain(s) - Medical Image Registration -

Teruyuki Yamasaki *

February 2022

This is a technical report on the 7th Brain(s) Contest[1], a data science competition held by FUJIFILM AI Academy Brain(s) from December 1, 2021 to January 17, 2022. Its topic is *medical image registration*, a key process of medical imaging systems. In this competition, my solutions get the first place in Q3 (and also in Q2 just in terms of the score). This paper is intended to share my solutions and experiences in this competition.

Contents

*Department of Biological Sciences, Graduate School of Science, the University of Tokyo

[1] https://fujifilmdatasciencechallnge.mystrikingly.com/

1 About Brain(s) Contests

1.1 Brain(s) Contests by FUJIFILM AI Acedemy Brain(s)

1.1.1 FUJIFILM AI Academy Brain(s)

FUJIFILM AI Academy Brain(s) was launched in October 2018 as a center for the development of more advanced, next-generation AI technologies to solve social issues through co-creation with academia. With the mission of "creating future AI technology by fusing people's wisdom with artificial intelligence (AI)", it has played a role as a research and development center for AI researchers of the Fujifilm Group and also as a place where they can carry out collaborative research with the leading academia for the purpose of bringing together their respective knowledge = Brain(s) and driving the development of AI technologies.

1.1.2 Brain(s) Contests

Fujifilm holds the Brain(s) Contests online in order to nurture and produce engineers who will contribute to the future development of the industry by encouraging young engineers to become more interested in artificial intelligence, machine learning and data science. Topics in the contests are related to the AI technologies of Fujifilm, such as imaging technologies and materials informatics. Students under 28 in any field regarding artificial intelligence, machine learning and data science can participate in the competitions, only as individuals.

1.2 The 7th Brain(s) Contest

1.2.1 Overview

The 7th Brain(s) Contest is about *medical imaging* in healthcare, one of the business fields of Fujifilm. To be more precise, it centers around *image registration*, a key technique in medical imaging systems, which allows for aligning multiple images into a single coordinate frame so that corresponding pixels represent homologous biological points[2].

Participants are required to solve three problems regarding image registration, Q1, Q2 and Q3: Q1 is intended as a tutorial exercise on affine transfomations, a key tool in image registration; in Q2 it is expected to perform image registration on a pair of two-dimensional MRI images with the same modality; Q3 is about image registration on a pair of multi-modal three-dimensional MRI images. The MRI images in Q2 and Q3 are obtained from IXI Dataset[3]. Performances in Q2 and Q3 are evaluated based on respective criteria.

[2] https://simpleelastix.readthedocs.io/Introduction.html#image-registration
[3] https://brain-development.org/ixi-dataset/

2 Q1: Affine Transformation Tutorial

2.1 Task

The goal of this task Q1 is to familiarize oneself with affine transformations, a key tool in image registration. Given a set of two-dimensional coordinates of source points, one has to compute the target points that are supposed to be obtained by applying to the source points the indicated series of affine transformations, which are composed of rotation, scaling, and translations.

2.2 Solution

The task Q1 is expected to be solved by using affine transformation matrices corresponding to the given procedures of two-dimensional transformations. Such matrices are obtained by multiplying rotation, scaling and translation matrices in appropriate ways, which are described in the following section.

2.2.1 Affine Transformation Matrices

An affine map, which is composed of a linear map and a translation, can be represented as:

$$f : \boldsymbol{x}_{src} \in \mathbb{R}^n \mapsto \boldsymbol{x}_{tgt} = A\boldsymbol{x}_{src} + \boldsymbol{b} \in \mathbb{R}^n,$$

where the multiplication by an invertible matrix $A \in \mathbb{R}^{n \times n}$ represents the linear map and the addition of a vector $\boldsymbol{b} \in \mathbb{R}^n$ the translation. By introducing *homogenous systems*, one can rewrite such an affine transformation using a single matrix multiplication:

$$\begin{pmatrix} \boldsymbol{x}_{tgt} \\ 1 \end{pmatrix} = \begin{pmatrix} A & \boldsymbol{b} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{src} \\ 1 \end{pmatrix}$$

The above augmented matrix is called an *affine transformation matrix*.

In two-dimensional coordinate systems, a general affine transformation can be written by using six parameters for the first two rows of the matrix:

$$\begin{pmatrix} x_{tgt} \\ y_{tgt} \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{src} \\ y_{src} \\ 1 \end{pmatrix}.$$

Affine transformations of rotation, scaling, and translation, which are transformations concerned in this task, are given as follows:

- **Rotation**:
  (counter-clockwise) rotation around the origin $O(0,0)$ by angle $\theta \in \mathbb{R}$

$$R(\theta) := \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **Scaling**:
  scaling around the origin $O(0,0)$ by factor $\lambda \in \mathbb{R}_{>0}$

$$S(\lambda) := \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **Translation**:
  translation by b_1 along x-axis and b_2 along y-axis with $b_1, b_2 \in \mathbb{R}$

$$T(b_1, b_2) := \begin{pmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \\ 0 & 0 & 1 \end{pmatrix}$$

It should be noted that, if one wants to perform rotation or scaling around a general (non-origin) point $P(p_1, p_2) \in \mathbb{R}^2$, it is necessary to translate the data points before and after the operation so that the center of the operation is placed at the origin during the rotation or scaling:

$$R'(\theta; p_1, p_2) = T(p_1, p_2)\, R(\theta)\, T(-p_1, -p_2)$$
$$S'(\lambda; p_1, p_2) = T(p_1, p_2)\, S(\lambda)\, T(-p_1, -p_2)$$

2.3 Results

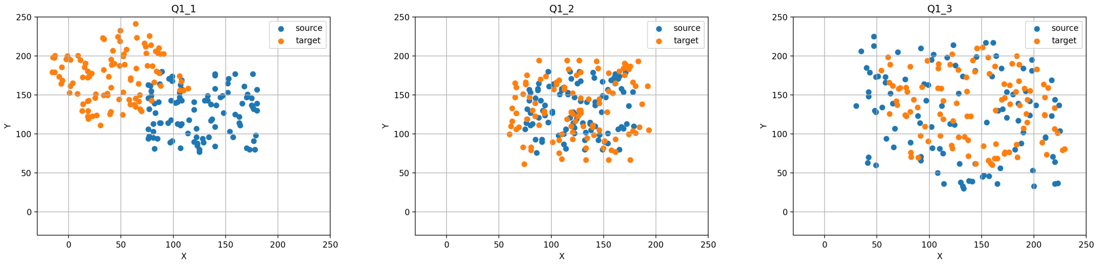

Figure 1: The results of the three tasks in Q1

3 Q2: Monomodal 2D Medical Image Registration

3.1 Task

The task Q2 is image registration on mono-modal two-dimensional medical images. Given a pair of source and target images, one has to estimate the unknown transformation that warps the source image to the target image. Both the source and target images are T1-weighted MRI images of the brain obtained in the following way: the source images by slicing the T1-weighted image volumes of IXI Dataset into two-dimensional images along the horizontal plane; the target images by applying to respective source images unknown spatial transformations, which should be recovered. An example of the provided 7986 image pairs is shown in 3.1

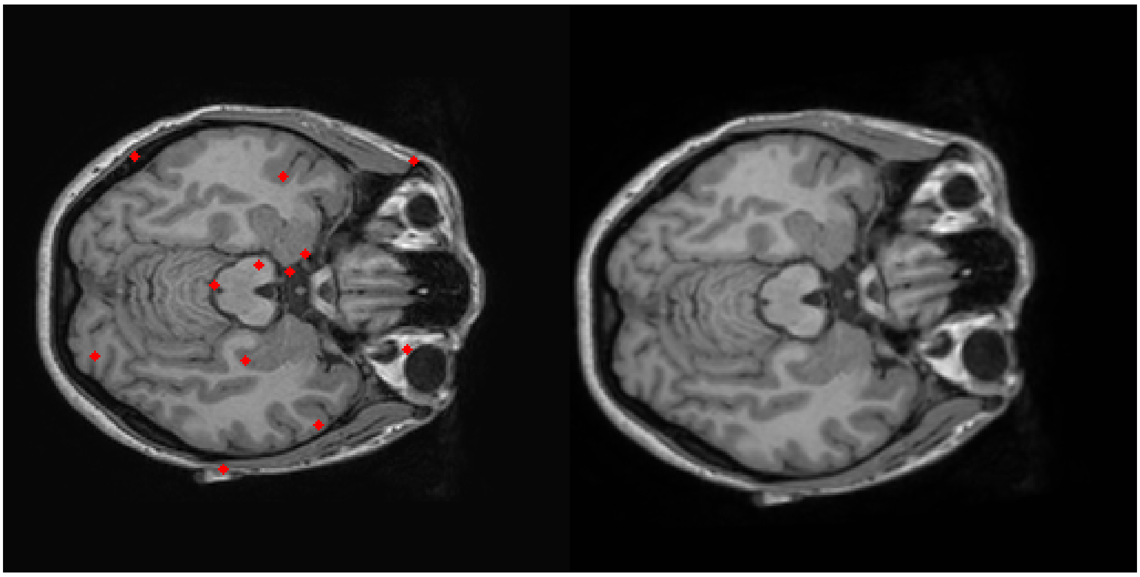

Figure 2: The source (left) image with key points and the target image (right) of IXI002-Guys-0828_051

To evaluate image registration performances a set of pixel coordinates of source key points are provided together with each source image, the estimated correspondences of which in the target image frame are supposed to be computed and compared to the ground truth. The mean Euclidean distance between the estimated target key points and the ground truth is used as the score.

3.2 Solution

My solution to Q2 is to perform image registration by concatenating the Euler transformation, the affine transformation, and the B-spline transformation, in a coarse-to-fine manner. In my implementation, though it did not violate rules of the competition, most of the optimization steps is outsourced by using ITKElastix[4], an ITK Python interface to elastix[5], which is an open source software for image registration. Therefore, I would like to give more space for description of Q3. The algorithm implemented using the library got the best score 0.856 in Q2, which was tied with others (the scores are rounded to the nearest thousandth). The final winner of Q2 showed an elegant solution using generative adversarial networks (GANs), awarded for his sophisticated solution and the smallest number of outliers among the tied best solutions.

Before using the ITKElastix library to tackle the non-linear part of deformations, I tested approaches that perform brute force template matching using the normalized cross-correlation (NCC): A template patch around each key point in the source image is extracted to be compared to the same size patch around every pixel in the target image; the pixel in the target image that gives the best NCC score is determined as the correspondence to each source key point. Such an approach works well to some extent, and gives its maximum score 2.056 when the patch radius is set to 9, but still involves not a few outliers. These outliers can be reduced by replacing them with the reprojected points by an affine transformation which is obtained by RANSAC or least squares methods. Such approaches improve the score up to 1.297, but still lack capability of capturing non-linear deformations and fail to eliminate obvious outliers.

3.3 Results

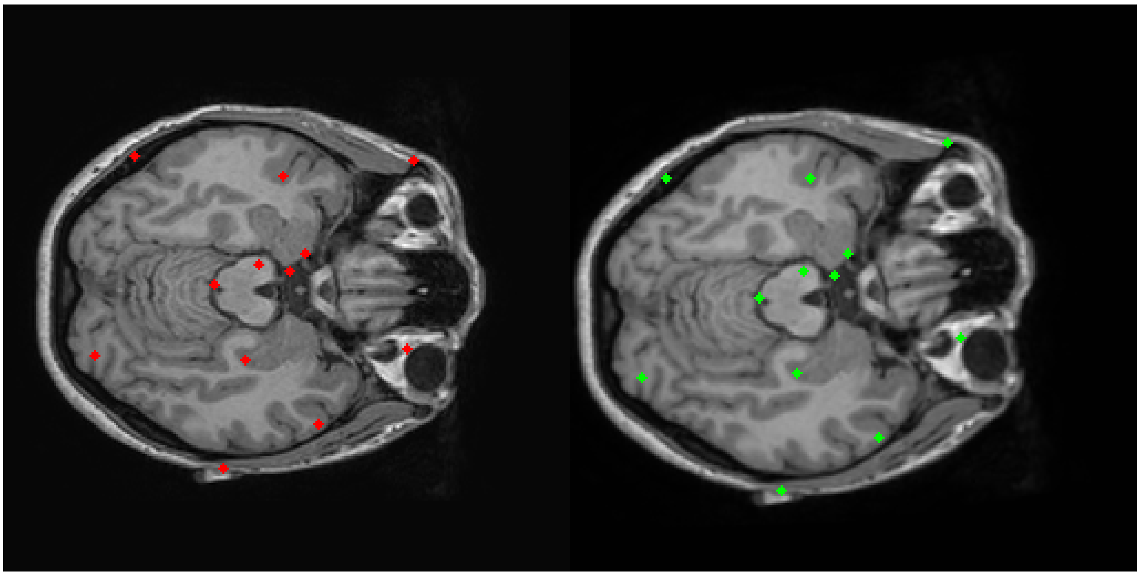

Figure 3: The source (left) and target (right) images with key points of IXI002-Guys-0828_051

[4]https://github.com/InsightSoftwareConsortium/ITKElastix
[5]https://elastix.lumc.nl/

4 Q3: Multimodal 3D Medical Image Registration

4.1 Task

The goal of this problem Q3 is to perform image registration on multi-modal three-dimensional medical images. The source images are T2-weighted, while the target images PD-weighted. Both types of images are obtained from the IXI Dataset. While in the original IXI Dataset each pair of T2 and PD-weighted images can be assumed to be registered beforehand, the target images in this task are given by warping the original PD-weighted volumes by unknown transformations, which should be recovered for each pair of volumes.

Estimations are evaluated just as in Q2: A set of voxel coordinates of multiple key points are provided together with each source volume, the corresponding voxel coordinates of which in the target volume have to be estimated. The scores are calculated as the mean Euclidean distance between the estimated target key points and the ground truth.

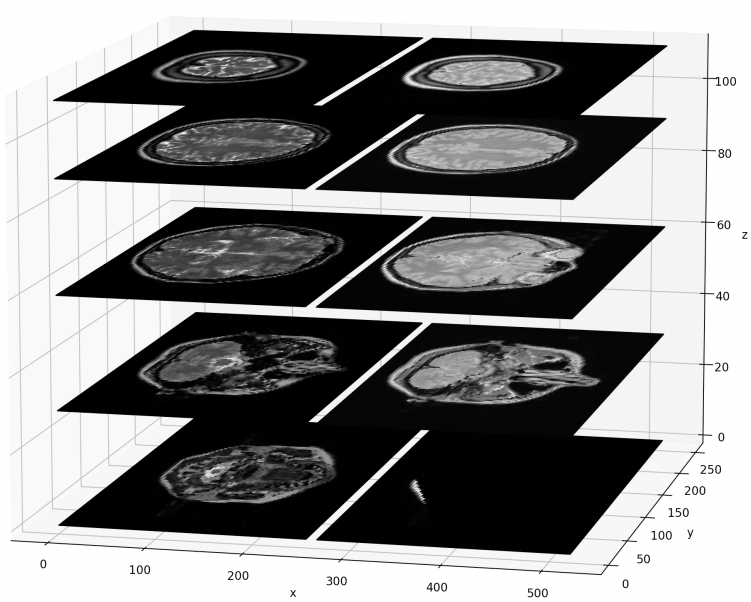

Figure 4: The source (T2; left) and target (PD; right) images of IXI002-Guys-0828

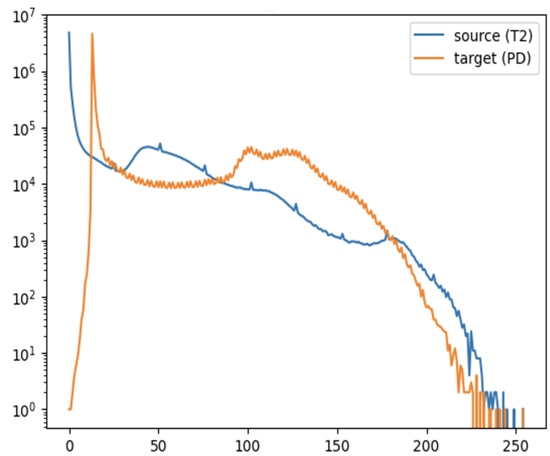

Figure 5: The histograms of the source (blue) and target (yellow) images of IXI002-Guys-0828

4.2 Solution

My approach to Q3 is to get z-stacked gradient magnitudes of the source and target volume images, respectively, and then minimize the sum of squared differences between them, the details of which are described in the following subsubsections. An overview of my solution is shown in Algorithm 1.

Algorithm 1 Solution to Q3 - Multimodal 3D Medical Image Registration

$SourceDataList \leftarrow$ readJson($path/to/keypoints\_source.json$)
$TargetDataList \leftarrow$ deepcopy($SourceDataList$)
$numDataSets \leftarrow$ length($SourceDataList$) ▷ 577
$factors \leftarrow (10, 3)$ ▷ factors for downsampling

for $i = 1 : numDataSets$ **do**
  $Id \leftarrow SourceDataList[i]['IXI\_ID']$
  $V_{src} \leftarrow$ readVolume($path/to/volumes\_source/\{Id\} * .npz$)
  $V_{tgt} \leftarrow$ readVolume($path/to/volumes\_target/\{Id\} * .npz$)

  $G_{src} \leftarrow$ computeZstackedGradientMagnitudes(V_{src})
  $G_{tgt} \leftarrow$ computeZstackedGradientMagnitudes(V_{tgt})

  $G_{src} \leftarrow$ normalize(G_{src})
  $G_{tgt} \leftarrow$ normalize(G_{tgt})

  $M \leftarrow$ IdentityMatrix($n = 4$) ▷ begin with the identity warp
  for $f \in factors$ **do** ▷ perform downsampling for efficiency
    $G_{src}\text{-}f \leftarrow$ downsample(G_{src}, f)
    $G_{tgt}\text{-}f \leftarrow$ downsample(G_{tgt}, f)
    $M \leftarrow$ rescaleTransform($M, 1/f$)
    $M \leftarrow \text{argmin}_M \sum \{G_{tgt}\text{-}f - \text{AffineTransform}(G_{src}\text{-}f; M)\}^2$ ▷ SSD minimization
    $M \leftarrow$ rescaleTransform(M, f)
  end for

  $X_{src} \leftarrow$ intoHomogenousCoordinates($SourceDataList[i]['keypoints']$)
  $X_{tgt} \leftarrow$ MatrixMultiplication(M, X_{src})
  $TargetDataList[i]['keypoints'] \leftarrow$ intoKeypointList(X_{tgt})
end for

saveJson($path/to/keypoints\_target.json, TargetDataList$)

4.2.1 Acquisition of Z-stacked Gradient Magnitudes

Since the source and target images have different modalities, naively applying the optimization methods to the image registration does not work well in this case. To overcome this problem, z-stacked gradient magnitudes of respective z-planes of the volumes can be used for comparison instead of the given volumes.

The algorithm to obtain z-stacked gradient magnitudes is shown in Algorithm 2. The derivatives of each z-plane image $I(z)$ are approximated by finite differences using convolutions with the so-called *Sobel* operators as follows:

$$
\begin{aligned}
I_x(z) &= \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} * I(z), \\
I_y(z) &= \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} * I(z)
\end{aligned}
$$

where $*$ here denotes the 2-dimensional signal processing convolution operation. Then, the gradient magnitude of the z-plane image is given by computing the L_2-norm of the image gradient $\nabla I(z) = (I_x, I_y)$ in a pixel-wise manner:

$$g(z) := \sqrt{I_x(z)^2 + I_y(z)^2}, \quad \text{for all } z = 1, 2, \cdots, z_{max}$$

By stacking all these gradient magnitudes $g(z), z = 1, 2, \cdots, z_{max}$, along the z-axis, into volume data, and normalizing them as shown in Algorithm 3, we get the z-stacked gradient magnitudes G, which can be more effective for the similarity comparison than the original volumes.

Algorithm 2 computeZstackedGradientMagnitudes

function COMPUTEZSTACKEDGRADIENTMAGNITUDES($volume$)
  $SobelFilter \leftarrow$ MatrixMultiplication($[1, 2, 1]^T[-1, 0, 1]$)
  $depth, height, width \leftarrow$ shape($volume$)
  $grads \leftarrow$ zeros(shape $= (depth, height, width)$)

  for $i = 1 : depth$ **do**
    $image \leftarrow volume[i]$ ▷ a $height \times width$ image
    $I_x \leftarrow$ convolve($SobelFilter, image$) ▷ image gradient along x-axis
    $I_y \leftarrow$ convolve($SobelFilter^T, image$) ▷ image gradient along y-axis
    $grads[i] \leftarrow \sqrt{I_x^2 + I_y^2}$ ▷ the gradient magnitude (pixelwise)
  end for

  return $grads$
end function

Algorithm 3 normalize

function NORMALIZE($volume$)
  $\delta \leftarrow 1e - 10$ ▷ to prevent division error
  $minV \leftarrow$ min($volume$) ▷ minimum voxel value
  $maxV \leftarrow$ max($volume$) ▷ maximum voxel value
  return $(volume - minV)/(maxV - minV + \delta)$ ▷ voxelwise
end function

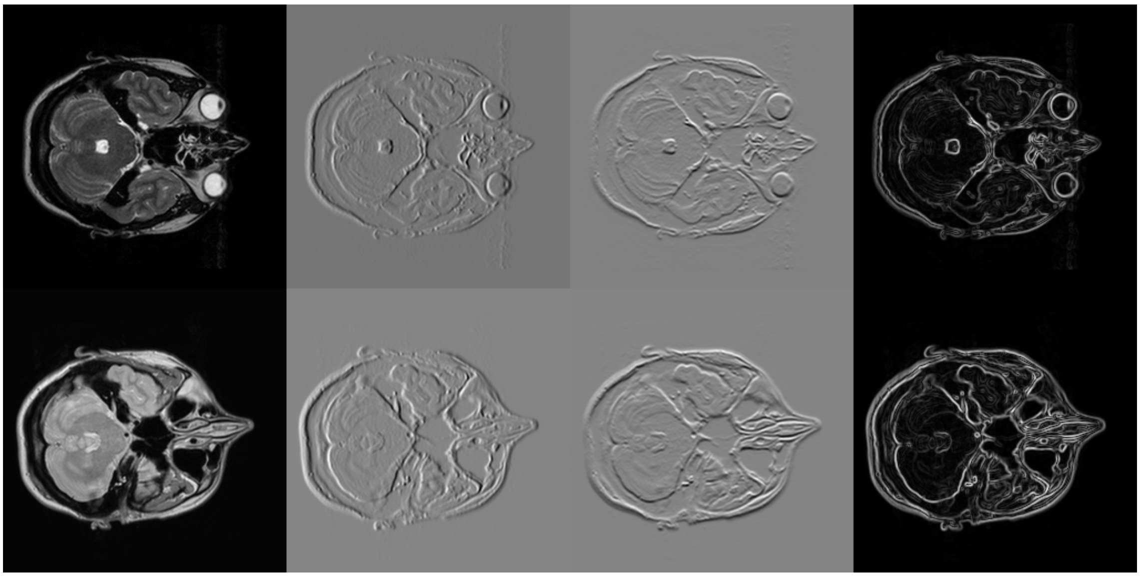

Figure 6: The origial, I_x, I_y, and gradient magnitude from left to right of the source (T2; top) and target (PD; bottom) images at $z = 40$ of IXI002-Guys-0828

In order to make the optimization more efficient and robust, the gradient magnitude volumes are downsampled into smaller volume data. Used in my implementation for this step is the *downscale_local_mean* method [6] of the *transform* module of the *scikit-image* library, which downsamples the N-dimensional input image by local averaging, i.e., calculating the local mean of elements in each block of size factors in the input image, which can contribute to removing high spatial frequency noise. Performing image registration on such multi-resolution pyramids in order from the lowest to the highest resolution is said to improve the capture range and robustness[7], paying more attentions to coarse structures in the beginning and to finer ones as the registration proceeds. While 3 resolutions are said to be sufficient in general, two factors $f = 10, 3$ are used for downsampling in my implementation, which are empirically determined for this task. Remember to rescale the translation parameters before and after optimization on each downsampled volume, because they are influenced by scale.

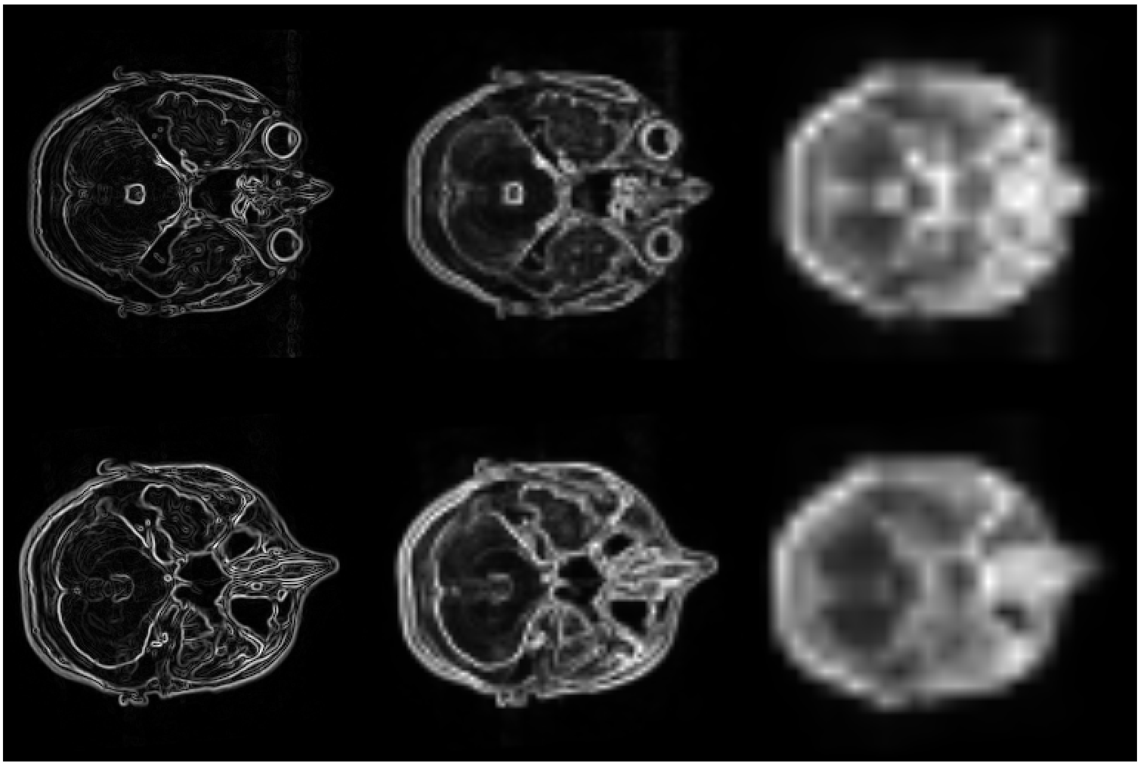

Figure 7: The downsapled images by factor $f = 1, 3, 10$ from left to right at $z = 40$ of IXI002

It should be noted that the z-stacked gradient magnitudes can be used to evaluate the similarity between the warped image and the fixed image, only on the assumption that the ground truth transformation does not involve significant rotational components other than around the z-axis or any other abnormally large deformations. Thankfully, those assumptions are likely to be met in medical imaging tasks, because medical datasets are usually obtained in carefully controlled settings. At least, in the given datasets for Q3, using the z-stacked gradient magnitudes seems to be as one of the effective ways for the similarity evaluation.

In addition to gradient magnitudes, approaches stacking the respective derivatives I_x, I_y and some feature scores such as Harris scores and Shi-Tomashi scores are also tested, only to be found ineffective in this case. The derivatives I_x and I_y themselves are variant to rotation, though they can derive similar intensity patterns from the source and target images with different modalities. A slight rotation of an image can cause significant difference in intensity patterns around edges and corners in its derivatives, which is not appropriate for image registration. On the other hand, feature scores such as Harris scores and Shi-Tomashi scores do give intensity patterns that are rotation-invariant, but the information of their intensity patterns seems too sparse to estimate the true gradient descent in the optimization step. Using the gradient magnitudes can contribute to

[6] https://scikit-image.org/docs/dev/api/skimage.transform.html#skimage.transform.downscale_local_mean
[7] https://simpleelastix.readthedocs.io/Introduction.html#image-pyramids

mitigating these shortcomings.

4.2.2 Minimization of the Sum of Squared Differences

Now that a pair of comparable image data is obtained, it is time to perform optimization to find a warp that aligns one to the other. For this step, a Gauss-Newton gradient descent non-linear optimization algorithm, which is called the Lucas-Kanade algorithm[8], is applied to minimize the sum of squared differences between the pair of image data.

Using the *least_squares* method of the *optimization* module of the *SciPy* library[9], a solver for non-linear least squares problems, outsources the implementation of the iterative optimization steps. Once a function which computes the vector of residuals and the initial parameters of the function are passed to the solver, it performs the minimization of the sum of squared residuals through gradient descent.

To gain a better understanding of what is computed inside, let us here take a look at the formulation of the optimization problem. It is derived as follows (Large part of the following description is based on a course material, which is provided for an exercise on Lukas-Kande tracker in Prof. Dr. Davide Scaramuzza's class of Vision Algorithms for Mobile Robotics Fall 2021[10] at the University of Zurich.):

Given an image $I : \boldsymbol{x} \in U \subset \mathbb{R}^n \mapsto y = I(\boldsymbol{x}) \in \mathbb{R}$, which maps n-dimensional coordinates to scalar intensity values, and a warp $W : (\boldsymbol{x}, \boldsymbol{p}) \in \mathbb{R}^n \times \mathbb{R}^m \mapsto \boldsymbol{y} = W(\boldsymbol{x}; \boldsymbol{p}) \in \mathbb{R}^n$, which is a function parameterized by m parameters $\boldsymbol{p}$ that transforms n-dimensional coordinates, $I_W := I(W(\boldsymbol{x}; \boldsymbol{p}))$ is said to be the image I warped by the warp W.

The goal of image registration can be formulated as to find a warp W that minimizes the sum of squared differences (SSD) E between the reference image I_R and a warped image I_W of the moving image I:

$$\begin{aligned} E(W) &:= \sum_{\boldsymbol{x} \in U} \left(I_W(\boldsymbol{x}) - I_R(\boldsymbol{x})\right)^2 \\ &= \sum_{\boldsymbol{x} \in U} \left(I(W(\boldsymbol{x}; \boldsymbol{p})) - I_R(\boldsymbol{x})\right)^2 \end{aligned}$$

This optimization method has the disadvantage that it has to explore a large space of possible W. The size of the space grows exponentially with the number of parameters m. However, thanks to the fact that $E(W)$ is mostly smooth and locally convex around the optimal solution W^*, we can use gradient descent to find W^* starting with an initial guess that can be assumed to be sufficiently close to W^*.

Thus, the optimization can be reformulated as finding an increment $\Delta\boldsymbol{p}$ to that initial guess $\boldsymbol{p}$ that minimizes the error E:

$$\Delta\boldsymbol{p}^* = \mathrm{argmin}_{\Delta\boldsymbol{p}} E$$

where

$$E := \sum_{\boldsymbol{x} \in U} \left(I(W(\boldsymbol{x}; \boldsymbol{p} + \Delta\boldsymbol{p}) - I_R(\boldsymbol{x}))\right)^2$$

That is, find $\Delta\boldsymbol{p}$ that equates the derivative of the error E with respect to $\Delta\boldsymbol{p}$ to zero:

$$\begin{aligned} E(\Delta\boldsymbol{p}) = \min_{\Delta\boldsymbol{p}} E \quad &\Leftrightarrow \quad \frac{\partial E}{\partial \Delta\boldsymbol{p}} = 0 \\ &\Leftrightarrow \quad \frac{\partial}{\partial \Delta\boldsymbol{p}} \sum_{\boldsymbol{x} \in U} \left(I(W(\boldsymbol{x}; \boldsymbol{p} + \Delta\boldsymbol{p})) - I_R(\boldsymbol{x})\right)^2 = 0 \end{aligned}$$

In order to find such $\Delta\boldsymbol{p}$, the steps of finding an increment $\Delta\boldsymbol{p}$ that approximately minimizes the error E and then updating the parameters with the found increment $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p}$ are iterated until the estimates of the parameters $\boldsymbol{p}$ converges.

[8] An Iterative Image Registration Technique with an Application to Stereo Vision. Bruce D. Lucas, Takeo Kanade https://www.researchgate.net/publication/215458777_An_Iterative_Image_Registration_Technique_with_an_Application_to_Stereo_Vision_IJCAI

[9] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html

[10] https://rpg.ifi.uzh.ch/teaching.html

The Gauss-Newton algorithm linearizes the non-linear expression of the error E by expanding it around $\boldsymbol{p}$ with a first-order Taylor approximation:

$$I(W(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p})) \sim I(W(\boldsymbol{x};\boldsymbol{p})) + \frac{\partial I(W(\boldsymbol{x};\boldsymbol{p}))}{\partial \boldsymbol{p}}\Delta\boldsymbol{p}$$

to yield

$$\frac{\partial}{\partial \Delta\boldsymbol{p}} \sum_{\boldsymbol{x}\in U} \left(I(W(\boldsymbol{x};\boldsymbol{p})) + \frac{\partial I(W(\boldsymbol{x};\boldsymbol{p}))}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} - I_R(\boldsymbol{x}) \right)^2 = 0$$

which can be regarded as a least squares problem and which has a closed form solution as follows.

This equation can be rewritten in a simpler form by vectorizing $I(W(\boldsymbol{x};\boldsymbol{p}))$, $I_R(\boldsymbol{x})$ and the square root in the above equation into $\boldsymbol{i}, \boldsymbol{i}_R$ and $\boldsymbol{e}$, respectively, where

$$\boldsymbol{e} := \boldsymbol{i} + \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} - \boldsymbol{i}_R$$

in order to get

$$\frac{\partial}{\partial \Delta\boldsymbol{p}} \boldsymbol{e}^T\boldsymbol{e} = 0$$

Applying the chain rule and the fact that $\frac{\partial}{\partial \Delta \boldsymbol{e}}\boldsymbol{e}^T\boldsymbol{e} = 2\boldsymbol{e}^T$ and $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{\Delta p}} = \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}$ (simply look at the coefficient of $\Delta\boldsymbol{p}$ in the equation that gives $\boldsymbol{e}$)

$$\begin{aligned} \frac{\partial}{\partial \Delta\boldsymbol{p}} \boldsymbol{e}^T\boldsymbol{e} &= \frac{\partial}{\partial \Delta\boldsymbol{p}} \boldsymbol{e}^T\boldsymbol{e}\ \frac{\partial \boldsymbol{e}}{\partial \Delta\boldsymbol{p}} \\ &= 2\boldsymbol{e}^T\ \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}} \end{aligned}$$

yields

$$\boldsymbol{e}^T\ \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}} = 0$$

Applying $(AB)^T = B^TA^T$, we get

$$\frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T \boldsymbol{e} = \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T \left(\boldsymbol{i} + \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} - \boldsymbol{i}_R \right) = \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} + \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T (\boldsymbol{i} - \boldsymbol{i}_R) = 0$$

Thus,

$$\Delta\boldsymbol{p} = -\left(\frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}} \right)^{-1} \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T (\boldsymbol{i} - \boldsymbol{i}_R)$$

$H = \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}^T \frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}}$ is the so-called *Hessian*.

Since $\boldsymbol{i}_R$ is constant,

$$\frac{\partial \boldsymbol{i}}{\partial \boldsymbol{p}} = \frac{\partial}{\partial \boldsymbol{p}}(\boldsymbol{i} - \boldsymbol{i}_R).$$

Let us describe $\boldsymbol{r} := \boldsymbol{i} - \boldsymbol{i}_R$, and we get

$$\Delta\boldsymbol{p} = -\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{p}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{p}} \right)^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{p}}^T \boldsymbol{r},$$

which means that in order to compute an approximated increment $\Delta\boldsymbol{p}$, we only need to get the residuals $\boldsymbol{r}$ and its Jacobian with respect to the parameter $\boldsymbol{p}$, which can be numerically approximated. Remember that the non-linear least squares solver of SciPy takes as input the residual $\boldsymbol{r}$ which is a function of $\boldsymbol{p}$ and the initial guess of $\boldsymbol{p}$.

By iteratively updating $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p}$ with this derivation, the paramter $\boldsymbol{p}$ is expected to converge to the local minimum such that the estimated warp aligns the source image to the target image.

For the iterative optimization, approximately it takes $\sim 10^1$ [sec] – a few seconds for the volumes downsampled by factor $f = 10$ and the rest for those downsampled by factor $f = 3$, with the number of iterations being $\sim 10^1$ for each resolution. In the data set labeled as *IXI002-Guys-0828*, for example, the SSD between the volumes downsampled by factor $f = 10$ reduces from 5.57×10^5 to 1.99×10^5 in 13 iterations, while the SSD between the volumes downsampled by factor $f = 3$ reduces from 2.04×10^7 to 1.40×10^7 in 13 iterations. Note that the order of the cost gets higher in finer optimization steps. This is simply because the residual vectors becomes larger in higher resolution, while the starting point of a finer optimization step is optimized through its previous coarser optimization steps.

4.3 Results

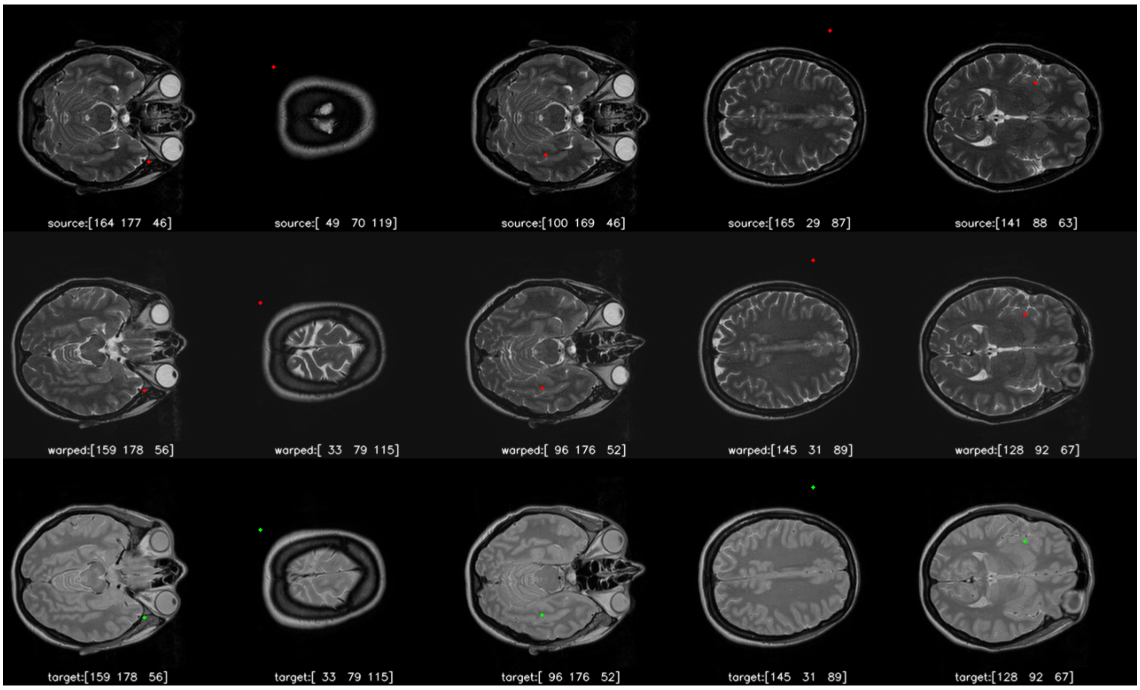

Figure 8: the source (T2;top), warped (T2;middle), and target (PD;bottom) images of five random key points in IXI002-Guys-0828

5 Summary

Using Python as a main language, I implemented the solutions to Q1, Q2 and Q3. In Q1, I familiarized myself with manipulation of affine transformation matrices and data format used in the competition. In Q2, although I relied largely on a package in implementing my final solution that uses a B-spline transform model, in prior attempts I tested many optimization methods such as template matching, linear least squares, and gradient descent, which itself became beneficial experience to deepen my understanding of computer vision algorithms and to improve my programming skills to implement them. In Q3, I managed to run the entire program to solve Q3, applying the Lucas-Kanade algorithm to the alignment between the volumes of image gradient magnitudes.

It is a great pleasure to get the best scores in both of Q2 and Q3 and to be awarded for the first place in Q3 after those efforts. In particular, I found it quite tough to implement the solution to Q3, because the visualization of results is not straightforward and the optimization is so heavy due to its large size of data and the parameter space. My program for Q3 did not work well until two days before the deadline of the competition. Also I wanted to try implementing my final solutions to Q2 without relying on the mentioned package and to test some improvements, if I had time after finishing Q3. In conclusion, it was a beneficial opportunity to learn medical imaging and to improve my data science skills through competing the scores, and sometimes sharing the knowledge, with other participants.

6 Acknowledgements

I would like to thank all those who concerned to the organization of the 7th Brain(s) Contest for providing us with such a valuable opportunity to test and improve our data science skills in imaging technologies through the well-structured problems, and to get ourselves connected with other students and professionals through discussing and sharing insights about the competition online. The knowledge, discussion, and sample codes that were shared there were beneficial for improving my scores and my understanding of the field. I appreciate the camera that I received for the prize, X Series FUJIFILM X100V, a Fujifilm's degital camera which has a compact body with elegant classic design and has one of the most sophisticated sensors and processors. I would like to use this camera to capture many unforgettable moments for a long time. Also I would like to thank other participants for working hard and competing together on the same problem, albeit individually, which kept me motivated to seek better solutions.

I also appreciate the insightful course on computer vision, Vision Algorithms for Mobile Robotics Fall 2021 by Prof. Dr. Davide Scaramuzza. Through the lectures and exercises in this course, I gained many insights and skills in computer vision and, though I was a beginner of this field before this class, I got much interested in it. Without the experience in this course, I would not have got the solid back ground enough to win this competition.

Lastly, I believe all these experiences and insights that I gained through this competition will be beneficial for my future work.

7 References

- Davide Scaramuzza, "Optical Flow and KLT Tracking" UZH-DINF2039 / ETH-151-0632-00L: Vision Algorithms for Mobile Robotics. Class lecture at the University of Zurich, Zurich, December 2, 2021, https://rpg.ifi.uzh.ch/teaching.html.
- Bruce D. Lucas and Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision." In Proceedings of the International Joint Conference on Artificial Intelligence, 1981.
- IXI Dataset, accessed February 18, 2022, https://brain-development.org/ixi-dataset/.
- "Introduction", SimpleElastix Documentation, accessed February 18, 2022, https://simpleelastix.readthedocs.io/Introduction.html.
- ITKElastix - an ITK Python interface to elastix, accessed February 18, 2022, https://github.com/InsightSoftwareConsortium/ITKElastix.