Medical Image Registration and Applications

Lab 1: Intensity Based Image Registration

Submitted By

Md. Kamrul Hasan

Fakrul Islam Tushar

Submitted To

Robert Martí Marly, PhD

robert.marti@udg.edu

October 25, 2018

Contents of the report		**Page No**

1. Introduction

Image registration is one of the prior steps for building computational model and Computer added diagnosis (CAD) which is the processes of transferring images into a common coordinate system, so that corresponding pixels represents homologous biological points [1]. In this lab, we have familiarized with the concepts and framework of image registration based on two different transformation techniques namely *"rigid transformation"* and *"affine transformation"* for brain MRI. Comparisons also have been accomplished for single-resolution and multi-resolution registration for the same images in both rigid transformation and affine transformation. Different quantitative and qualitative matric performance are also been observed for all the experiments.

2. Registration framework

Fig.1 shows the typical framework for image registration which have two input images, one is termed as fixed image and another one is termed as moving image. A transform represents the spatial mapping of points from the fixed image to points in the moving image. The metric is a key component in the registration process. It uses information from the fixed and moving image to compute a similarity value. The derivative of this value tells us in which direction we should move the moving image for better alignment. The moving image is moved in small steps, and process is repeated until a convergence-criteria has satisfied.

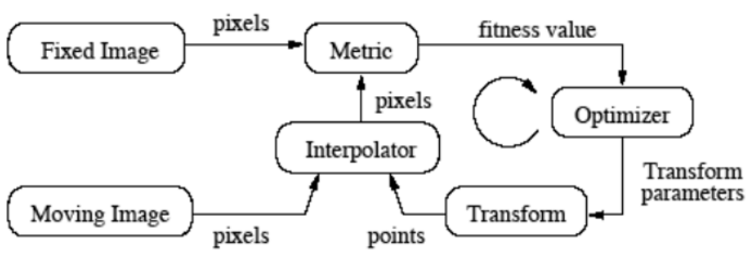


Fig. 1: Typical framework for image registration [2].

3. Similarity metric

Similarity metric can be defined as a measurement that quantifies the degree of similarity between intensity patterns in two images [3]. The selection of similarity metric influentially depends on the modality of the images to be registered. In our experiments, we have used three different metrics that are briefly described below-

3.1 Normalized Cross-correlation (NCC)

Normalized Cross-correlation (NCC) computes pixel-wise cross-correlation and normalized by the square root of the autocorrelation of the images which is invariant to linear differences between intensity distributions. Mathematically, NCC can be expressed as below-

$$NCC\left(I_{fixed}, I_{moving}\right) = \frac{\sum_{x=0}^{X}\sum_{y=0}^{Y}\left(I_{fixed}(x,y) - \overline{I_{fixed}}\right) \times \left(I_{moving}(x,y) - \overline{I_{moving}}\right)}{\sqrt{\sum_{x=0}^{X}\sum_{y=0}^{Y}\left(I_{fixed}(x,y) - \overline{I_{fixed}}\right)^2 \times \sum_{x=0}^{X}\sum_{y=0}^{Y}\left(I_{moving}(x,y) - \overline{I_{moving}}\right)^2}}$$

3.2 Sum of Squared Distance (SSD)

Sum of squared differences (SSD) is one of measure of match that based on pixel by pixel intensity differences between the fixed and moving images which calculates the summation of squared for the product of pixels subtraction between two images and normalized by total number of pixels in the image. Mathematically, SSD can be expressed as below-

$$SSD\left(I_{fixed}, I_{moving}\right) = \frac{\left(\sum_{x=0}^{X}\sum_{y=0}^{Y}\left(I_{fixed}(x,y) - I_{moving}(x,y)\right)^2\right)}{Total\ numbers\ of\ pixels}$$

3.3 Gradient Normalized Cross-correlation (GNCC)

Gradient Normalized Cross-correlation (GNCC) computes a non-linear comparison of gradient structure in overlapping image regions and offers intrinsic invariance to changing illumination, without recourse to background-model adaptation [4]. Mathematically, GNCC can be expressed as below-

$$GNCC\left(I_{fixed}, I_{moving}\right) = \frac{\sum_{xy}\left(\frac{\partial I_{fixed}(x,y)}{\partial x} \times \frac{\partial I_{moving}(x,y)}{\partial x} + \frac{\partial I_{fixed}(x,y)}{\partial y} \times \frac{\partial I_{moving}(x,y)}{\partial y}\right)}{\sqrt{\sum_{xy}\left(\frac{\partial I_{fixed}(x,y)^2}{\partial x} + \frac{\partial I_{fixed}(x,y)^2}{\partial y}\right)} \times \sum_{xy}\left(\frac{\partial I_{moving}(x,y)^2}{\partial x} + \frac{\partial I_{moving}(x,y)^2}{\partial y}\right)}$$

In case of the brain4 for all the experiments, the X and Y directional gradient of moving image are multiplied by the -1. The reason for this multiplication is shown in Fig. 2. The intensity of the background for brain4 and brain3 is opposite from each other. Fig. 2(a) and 2(b), are the Y directional gradient for brain3 and brain4 respectively. From the Fig. 2(b), it is seen that Y directional gradient of brain4 has opposite response than Y directional gradient of brain3. But, this problem can be solved if we multiply Y directional gradient of brain4 by -1 (see red circled part from Fig. 2.). So, all the cases for brain4 to be registered there is different GNCC function (*Gradient_cc_2.m*) in code folder.

(a) (b) (c)

Fig. 2: Y direction gradient of brain3 (a), Y direction, and Negative of Y direction gradient of brain4 (b & c)

4. Transformation

Transformation maps coordinates from one coordinate system to another and is often described by its Degrees of Freedom (DOF), which is the number of independent ways that the transformation can be happened. In our experiments, we have used two different transformations (3 DOF and 7 DOF) that are briefly described below-

4.1 Rigid Transformation

A rigid transformation can register images that are related to rotation and translation. Rigid registration is one of the simplest of methods for linear transformation and often used as initialization for affine- and non-rigid transforms.

$$Rigid\ Trasnformation\ matrix = \begin{bmatrix} x\cos\theta & y\sin\theta & T_x \\ -x\sin\theta & y\cos\theta & T_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$Where,\ T_x = Translation\ along\ x\ axis\ \ T_y = Translation\ along\ y\ axis,\ and\ \theta = Rotation$$

Above equation represents the rigid transformation matrix for the rigid transformation in 2D image registration which is controlled by 3 parameters (called 3-DOF) namely $T_x, T_{y,} and\ \theta$.

4.2 Affine Transformation

To perform affine registration, sharing and scaling is added to the rotation and translation of rigid transformation.

Affine registration= Rotation +Translation + Sharing+ Scaling= Rigid transformation + Sharing + Scaling

$$\text{Affine Transformation matrix} = \underbrace{\begin{bmatrix} x\cos\theta & y\sin\theta & T_x \\ -x\sin\theta & y\cos\theta & T_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Rotation \& Translation}} \times \underbrace{\begin{bmatrix} 1 & V_y & 0 \\ V_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Shearing}} \times \underbrace{\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Scaling}}$$

Where, $V_x =$ Shearing along x axis , $V_y =$ Shearing along y axis , $S_x =$ Scaling along x axis , $S_y =$ Scaling along y axis. To process affine registration in 2D, these 7 parameters (7 DOF) $[T_x, T_{y,}, \theta, V_x, V_y, S_x S_y]$ need to be scaled with the preferred weight by the scaling vector of same size.

Scale vector: In both cases, either rigid or affine transformation, during the implementation there is a scale vector which is used to emphasis (assign more weight) to some specific parameters that play crucial roles for the perfect registration. The selection of scale vector depends on the types (geometric orientation) of the images to be registered.

Another point needs to be mentioned that all the rotations of the image transformation for the moving image occurre at the center of the origin of the image coordinate which is $(0,0)$.

5. Single-resolution VS Multi-resolution

Multi-resolution mean representing or analyzing image at more than one resolution. The idea is image pyramid which is started with the original image and shrink it half of the size after gaussian filtering, and again shrined by same way and at the top of the pyramid, we will get image having size 1×1. So, the advantage of multi-resolution over single-resolution is that the size of the object is smaller in sub-sampled image and it improves the capture range and robustness of the registration. In our multi-resolution experiments, registration has started with the low resolution and cumulatively increase the resolution until original image. The parameters obtained in the lower resolution has been used an initialization of the succeeded higher resolution. The typical gaussian pyramid and different level of gaussian pyramid images are shown in Fig. 3. As the level of gaussian pyramid increases, the image has lower resolution.

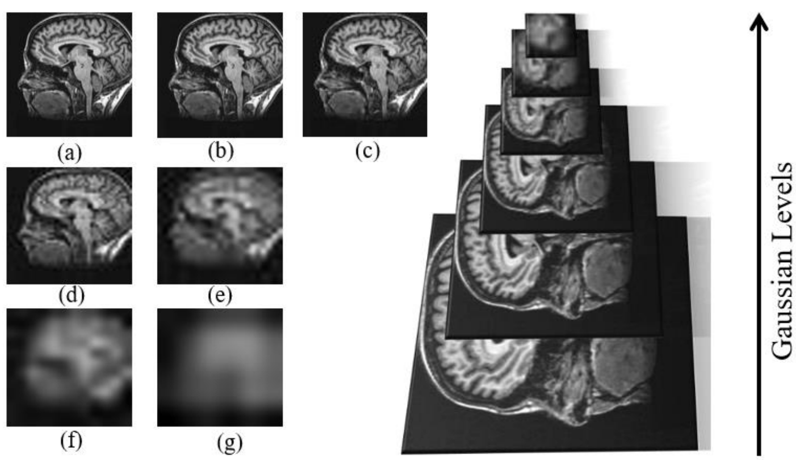
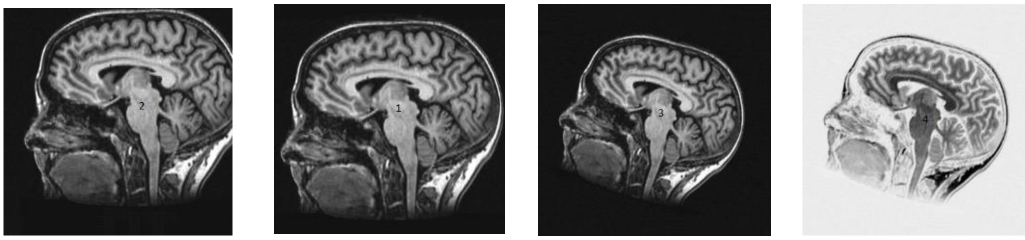


Fig. 3: Multi-resolution gaussian pyramid (right) and (left) image resolution after each level of gaussian a) original (Level=0), b) Level=1, c) Level=2, d) Level=3, e) Level=4, f) Level=5, and g) Level=6.

6. Results and Discussions

For this lab work, there are four images as shown in Fig. 4. In every case, Fig. 4 (a) is the fixed image and others three Fig. 4 (b), Fig. 4 (c) and Fig. 4 (d) are moving images. Objectives are to get best parameters e.g. “Translation”, “Scaling”, “Rotation” and “Shearing” that align geometrically all the moving images to fixed image. In all the experiments, some hyper-parameters are kept constant to compare the results with each other’s. For examples, in optimizer maximum number of iterations was 1500, termination tolerance on the function was 10^{-30}, termination tolerance on parameters was 10^{-30} and maximum number of function evaluations was $2000 \times length(parameters)$. Those hyper-parameters have great influence on the computation time for the registration process. In our case, tolerance is very less and iteration is high. Although, no experiments have been done to select best hyper-parameters and optimize the computation time. But, in all the experiments, they are kept constant. In future, those parameters can be optimized using well known algorithm ***"grid search"*** to reduce the computation time.



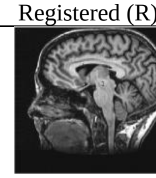
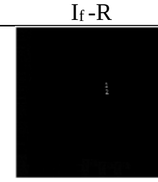
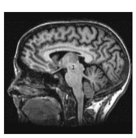
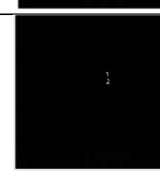
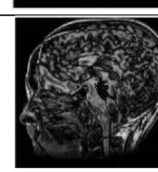
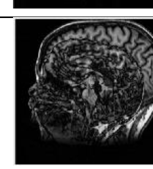
(a) (b) (c) (d)

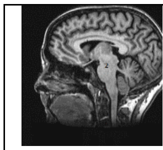
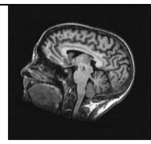
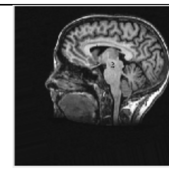
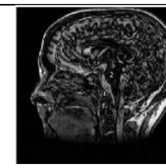
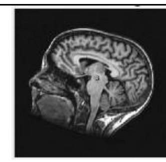
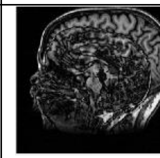
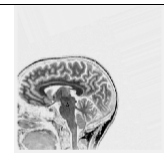
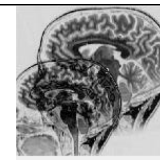
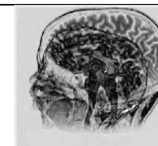
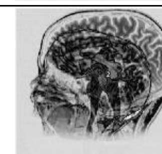
Fig. 4: Given fixed and moving images for the registration a) fixed, b), c), d) moving image

6.1 Results for Rigid Transformation with Single-resolution

For this experiment, in the MATLAB source code named *"Image_Registration.m"*, need to select ***Level_of_G_Pyramid=0*** where 0 means no gaussian down-sampling and registration will be done using only original images. Initial parameters for this experiment are $x = [0\ 0\ 0]$ with the scaling vector $x = [0.1\ \ 0.1\ \ 1]$. From the Table 1, it is seen that registration between brain1 (moving) and brain2 (fixed) for SSD and NCC metric, working with high degree performances (in both quantitively and qualitatively). Registration between brain1 (moving) and brain2 (fixed) for GNCC metric, is not working and possible reason for that might be due to bad initialization. It will be clearer after multi resolution, *weather it will working or not?* Since, in rigid transformation, only 3 DOF is possible, so registration between brain3 (moving) and brain2 (fixed) or brain4 (moving) and brain2 (fixed) will not work perfectly (due to not having scaling parameters). But, in case of NCC metric, it seen that model was able to rotate and translate which are not enough to register brain3 (moving) and brain2 (fixed) or brain4 (moving) and brain2 (fixed).

Table 1: Results for rigid transformation with single-resolution

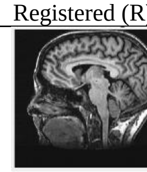
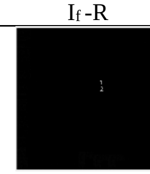
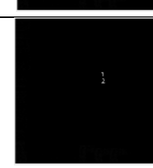
Fixed (I_f)	Moving	Metric	Parameters	Performances	Registered (R)	I_f-R
		SSD	$T_x = 23.98$ $T_y = -21.99$ $\theta = 0.0000^c$	SSD $= 2 \times 10^{-7}$ $Time = 40.08\ s$		
	 brain1.png	NCC	$T_x =$ 23.99 $T_y =$ -22.01 $\theta = 0.0000^c$	NCC = 0.9982 $Time = 40.97\ s$		
		Gradient NCC (GNCC)	$T_x = -0.636$ $T_y =$ 1.7838 $\theta = 0.0043^c$	GNCC = 0.0311 $Time = 43.48\ s$		
		SSD	$T_x =$ 0.5380 $T_y = -0.57$ $\theta = 0.0002^c$	SSD = 0.0575 $Time = 23.15\ s$		

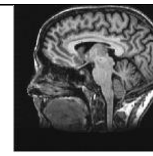
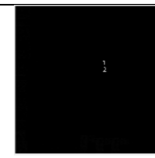
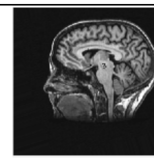
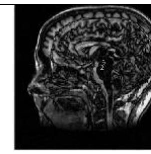
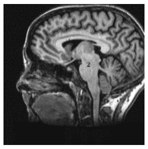
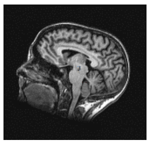
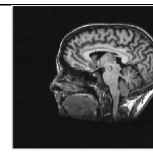
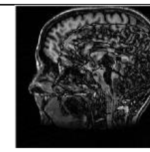
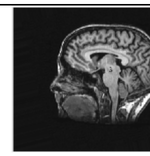
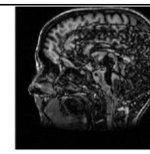
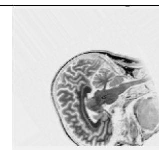
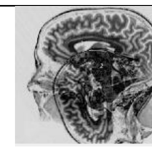
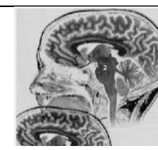
 brain2.png	 brain3.png	NCC	$T_x = 25.70$ $T_y = -39.86$ $\theta = 0.2260^c$	NCC = 0.5048 $Time = 37.35\ s$		
		Gradient NCC (GNCC)	$T_x = 0.400$ $T_y = -0.63$ $\theta = 0.0011^c$	GNCC = 0.0338 $Time = 28.97\ s$		
	 brain4.png	SSD	$T_x = -45.6$ $T_y = 82.70$ $\theta = 0.3487^c$	SSD = 0.3441 $Time = 52.21\ s$		
		NCC	$T_x = 0.000$ $T_y = 0.000$ $\theta = 0.0000^c$	NCC = 0.4059 $Time = 18.76\ s$		
		Gradient NCC (GNCC)	$T_x = 0.393$ $T_y = -0.71$ $\theta = 0.0012^c$	GNCC = 0.0315 $Time = 26.47\ s$		

6.2 Results for Rigid Transformation with Multi-resolution

For this experiment, in the MATLAB source code named *"Image_Registration.m"*, anyone need to select ***Level_of_G_Pyramid=6*** where, 6 means six times gaussian down-sampling and registration will be done using low resolution and take the initial parameters for the cumulative higher resolution. The initial parameters and scaling vector for this experiment also same as the previous experiment but after each low resolution registration only translation parameters again scaled double which is $scaleTxTy = [Tx = 2,\ Ty = 2,\ 1]$. From Table 2, registration between brain1 (moving) and brain2 (fixed) for GNCC metric, is perfectly working which was not working in the single resolution. *So, multi-resolution is better to solve the bad initialization.* As well as, for brain3 (moving) and brain2 (fixed), all the metrics can rotate and translate due to better finalization (see difference in Table 1 and Table 2). But, for brain4 (moving) and brain2 (fixed), still not working using rigid transformation either original image or multi-resolution for all the metrics. Now, for the brain4 to be registered with brain2, expected result may be obtained for the affine transformation.

Table 2: Results for rigid transformation with multi-resolution

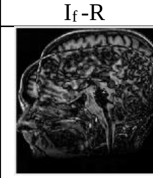
Fixed (I_f)	Moving	Metric	Parameters	Performances	Registered (R)	I_f-R
	 brain1.png	SSD	$T_x = 23.98$ $T_y = -21.99$ $\theta = 0.0000^c$	SSD = 0.0002 $Time$ $= 178.15\ s$		
		NCC	T_x =23.98 $T_y = -21.99$ $\theta = 0.0000^c$	NCC = 0.9982 $Time = 42.87\ s$		

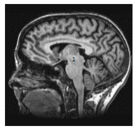
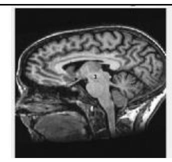
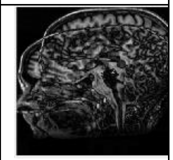
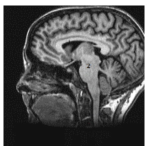
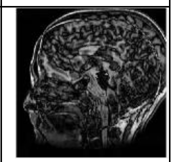
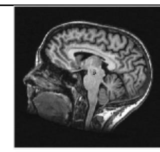
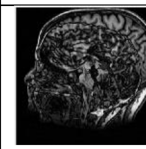
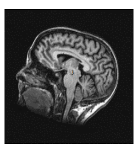
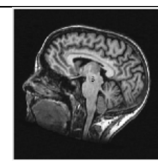
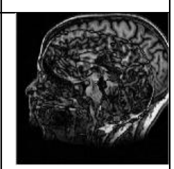
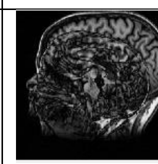
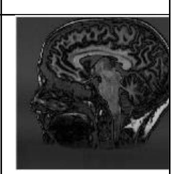
		Gradient NCC (GNCC)	$T_x = 23.9854$ $T_y = -21.99$ $\theta = 0.0000^c$	GNCC = 0.9886 $Time = 52.87\ s$		
		SSD	$T_x = 12.96$ $T_y = -40.65$ $\theta = 0.3019^c$	SSD = 0.0498 $Time = 77.91\ s$		
 brain2.png		NCC	$T_x = -3.82$ $T_y = -53.80$ $\theta = 0.2636^c$	NCC = 0.5189 $Time = 211.99\ s$		
	brain3.png	Gradient NCC (GNCC)	$T_x = -1.365$ $T_y = -56.43$ $\theta = 0.2356^c$	GNCC = 0.0638 $Time = 266.83\ s$		
		SSD	$T_x = -7.062$ $T_y = 75.096$ $\theta = 2.1818^c$	SSD = 0.3523 $Time = 57.28\ s$		
		NCC	$T_x = -186.4$ $T_y = 108.66$ $\theta = 0.1427^c$	NCC = 0.2973 $Time = 75.49\ s$		
	brain4.png	Gradient NCC (GNCC)	$T_x = 20.7280$ $T_y = -48.42$ $\theta = 0.6026^c$	GNCC = 0.0366 $Time = 164.68\ s$		

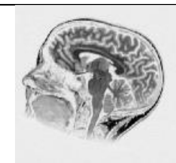
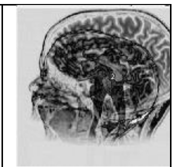
6.3 Results for Affine Transformation with Single-resolution

For this experiment, in the MATLAB source code named *"Image_Registration.m"*, anyone need to select ***Level_of_G_Pyramid=0*** where 0 means no gaussian down-sampling and registration will be done using only original images. Initial parameters for this experiment are $x = [0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0]$ with the scaling vector $x = [1\ \ 1\ \ 0.1\ \ 1\ \ 1\ \ 0.001\ \ 0.001]$. From the Table 3, is visible that both quantitive and qualitative results are bad for affine transformation with single resolution. In affine transformation, there is 7 DOF, so it is very crucial to initialize the parameters. In single resolution, sophisticated and clever initialization might solve the problems of bad registration in this experiment.

Table 3: Results for affine transformation with single-resolution

Fixed (I_f)	Moving	Metric	Parameters	Performances	Registered (R)	I_f-R
		SSD	$T_x = 0.47$ $T_y = 0.72$ $\theta = 0.12^c$ $S_x = 1.16$ $S_y = 0.89$ $V_x = -0.0001$	SSD = 0.0510 $Time = 83.30\ s$		

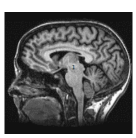
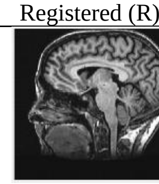
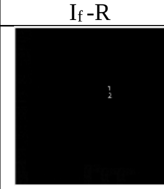
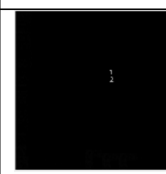
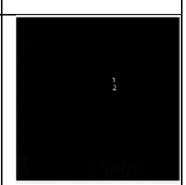
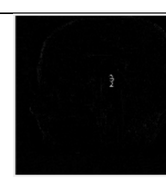
			$V_y = 6.9\ e - 5$			
	 brain1.png	NCC	$T_x = 0.920$ $T_y = 0.780$ $\theta = 0.120^c$ $S_x = 1.170$ $S_y = 0.880$ $V_x = -0.0001$ $V_y = -1.5\ e - 4$	NCC = 0.4815 $Time = 79.38\ s$		
 brain2.png		Gradient NCC (GNCC)	$T_x = 0.016$ $T_y = 0.008$ $\theta = -0.002^c$ $S_x = 0.990$ $S_y = 1.040$ $V_x = 0.000$ $V_y = -2.4\ e - 6$	GNCC = 0.0108 $Time = 59.74\ s$		
		SSD	T_x =-0.153 $T_y = 0.182$ $\theta = 0.0501^c$ $S_x = 0.916$ $S_y = 0.931$ $V_x = -0.00$ $V_y = -3.2e - 5$	SSD = 0.0552 $Time$ $= 54.60\ s$		
	 brain3.png	NCC	$T_x = 0.166$ $T_y = 0.757$ $\theta = 0.039^c$ $S_x = 0.929$ $S_y = 0.935$ $V_x = -0.00$ $V_y = -7.6e - 5$	NCC = 0.4330 $Time = 92.50\ s$		
		Gradient NCC (GNCC)	$T_x = -0.087$ $T_y = 0.1096$ $\theta = -0.011^c$ $S_x = 1.0047$ $S_y = 0.9660$ $V_x = 0.0000$ $V_y = 2.1\ e - 5$	GNCC = 0.0429 $Time = 68.95\ s$		
		SSD	$T_x = -0.03$ $T_y = -0.57$ $\theta = 0.0027^c$ $S_x = -0.045$ $S_y = 0.0010$ $V_x = 0.0000$ $V_y = 1.6e - 5$	SSD = 0.0435 $Time$ $= 73.40\ s$		
	 brain4.png	NCC	$T_x = 2.332$ $T_y = -2.74$ $\theta = -0.131^c$ $S_x = 443.2$ $S_y = 790.7$ $V_x = -0.0004$ $V_y = 3.6e - 6$	NCC = 0.3545 $Time = 79.49\ s$		

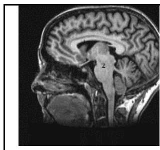
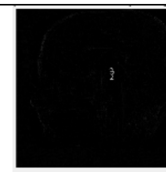
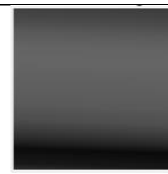
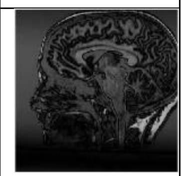
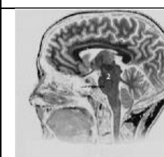
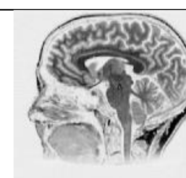
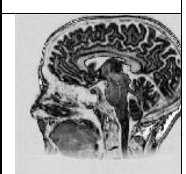
		Gradient NCC (GNCC)	$T_x = 0.043$ $T_y = 0.684$ $\theta = -0.005^{c}$ $S_x = 1.005$ $S_y = 0.964$ $V_x = -0.0001$ $V_y = -3.3e-5$	GNCC = 0.0426 $Time$ $= 114.80\ s$		

6.4 Results for Affine Transformation with Multi-resolution

For this experiment, in the MATLAB source code named *"Image_Registration.m",* anyone need to select ***Level_of_G_Pyramid=6*** where, 6 means six times gaussian down-sampling and registration will be done using low resolution and take the initial parameters for the cumulative higher resolution. The initial parameters and scaling vector for this experiment also same as the previous experiment but after each low resolution registration only translation parameters again scaled double which is $scaleTxTy = [Tx = 2,\ Ty = 2,\ 1,\ 1,\ 1,\ 1,\ 1]$. From the Table 4, is visible that both quantitative and qualitative results are better for affine transformation in multi-resolution compared to results of single resolution in Table 3 in case of brain2 (fixed) and brain1, brain3 (fixed) registration. In this case, both the quantitative and qualitative results are perfect. One of the reasons of the performance improvement in multi-resolution is that the size of sub-sampled image are smaller and avoiding the optimal initialization problem by getting the initial parameter from cumulation low resolution registration. In the case of brain4 as like previous cases affine multiresolution is also failed. It's because the intensity difference between the brain2 and brain4 is so high that matrixes are failed to converge towards the optimal values through optimizer.

Table 4: Results for affine transformation with multi-resolution

Fixed (I_f)	Moving	Metric	Parameters	Performances	Registered (R)	I_f-R
	 brain1.png	SSD	$T_x = 23.98$ $T_y = -21.99$ $\theta = 0.0248$ $S_x = 0.9996$ $S_y = 0.999$ $V_x = -0.025$ $V_y = 2.5e-2$	SSD $= 0.0002$ $Time$ $= 335.05\ s$		
		NCC	$T_x = 23.98$ $T_y = -21.99$ $\theta = 0.0000$ $S_x = 0.9999$ $S_y = 1.0000$ $V_x = -0.000$ $V_y = 4.1e-5$	NCC $= 0.9982$ $Time$ $= 196.41\ s$		
		Gradient NCC (GNCC)	$T_x = 23.984$ $T_y = -22.01$ $\theta = 0.0005$ $S_x = 1.0000$ $S_y = 1.0000$ $V_x = -0.0005$ $V_y = 5.6e-4$	GNCC $= 0.9886$ $Time$ $= 228.06\ s$		
		SSD	$T_x = 9.1933$ $T_y = -27.307$ $\theta = 0.2266$ $S_x = 0.7878$ $S_y = 0.7879$	SSD = 0.0003 $Time$ $= 318.54$		

 brain2.png	 brain3.png		$V_x = 0.0158$ $V_y = -1.6e-2$			
		NCC	$T_x = 9.1952$ $T_y = -27.308$ $\theta = 0.2244$ $S_x = 0.7881$ $S_y = 0.7881$ $V_x = 0.0186$ $V_y = -1.9e-2$	NCC $= 0.9968$ $Time$ $= 161.38\ s$		
		Gradient NCC (GNCC)	$T_x = 9.1945$ $T_y = -27.31$ $\theta = 0.2435$ $S_x = 0.7860$ $S_y = 0.7861$ $V_x = -0.0055$ V_y =5.5e-3	GNCC $= 0.9774$ $Time$ $= 231.77\ s$		
		SSD	$T_x = -6.7328$ $T_y = 7.4843$ $\theta = -0.0000$ $S_x = 0.0227$ $S_y = 0.0000$ $V_x = -0.0437$ $V_y = -3.1e2$	SSD $= 0.0371$ $Time$ $= 390.92\ s$		
	 brain4.png	NCC	$T_x = 39.4414$ $T_y = 0.0002$ $\theta = 0.0334$ S_x =43.4973 $S_y = 5.1665$ $V_x = -0.0006$ $V_y = -9.4e-5$	NCC $= 0.0743$ $Time$ $= 282.12\ s$		
		Gradient NCC (GNCC)	$T_x = 9.1998$ $T_y = -27.31$ $\theta = 0.2469$ $S_x = 0.7856$ $S_y = 0.7856$ $V_x = -0.0099$ $V_y = 9.97e-03$	GNCC $= 0.9831$ $Time$ $= 180.59$		

References

[1] Viergever, M., Maintz, J., Klein, S., Murphy, K., Staring, M. and Pluim, J. "A survey of medical image registration – under review". *Medical Image Analysis*, 33, pp.140-144, 2016.

[2] Marti, Robert. "MIRA: Week 1: Intensity based Image registration", pp. 5, October 2018.

[3] Barbara Zitová, Jan Flusser. "Image registration methods: a survey", Image Vision Comput. 21 (11): 977-1000 (2003).

[4] O'Callaghan, Robert & Haga, Tetsuji. (2007). "Robust Change-Detection by Normalized Gradient-Correlation", DOI: 10.1109/CVPR.2007.383516.