Medical Image Registration and Applications Lab 1 Report

Kazi Saeed Alam, Gonzalo Esteban Mosquera Rojas
saeed.utsha@gmail.com, gonzalo.mosquerarojas2000@gmail.com
Erasmus Mundus Joint Master's Degree in Medical Imaging and Applications
University of Girona

I. Introduction and Problem Definition

Image Registration can be defined as the process of transforming a set of images into one single coordinate system. The image used as base is known as "fixed image", whereas the remaining one is called "moving image". In other words, the idea behind image registration is finding correspondence between points of two images of the same scene, in which one of the two has suffered some kind of transformation. This procedure is highly relevant in the field of medical imaging, since there are a lot of different imaging techniques (MRI, CT, PET, SPECT, etc), which are all acquired with different equipment, sensors, depths and points of view. Each one of this image modalities has its advantages and drawbacks. There might be cases in clinical practice where both modalities provide important but insufficient information as a standalone resource. For instance, one technique might be very good at the imaging of the brain structure but does not allow to see important lesions clearly, while the other can be really sensitive for detecting tumors but does not provide a general view of the brain anatomical structure. In such cases, image registration emerges as an essential tool to transform the images into one single coordinate system and therefore being able to take advantage of their key properties with the purpose of generating more reliable and accurate diagnosis of the patients.

During this lab session the problem of Image Registration was tackled. The primary goal was to understand the main concepts and steps involved in an image registration framework and being able to modify it to incorporate two similarity metrics for the evaluation of the results, namely, normalised cross-correlation and normalised gradient cross-correlation. The framework was also edited to include affine transformations and to support multi-resolution registration. Once the code was modified to add the new requirements, the experimental part was about performing the registration of brain images from four different patients with different combinations of images, evaluation metrics and levels of resolution. The analysis consisted on finding the main differences in the results according to the parameters combination, the metric values as a function of the number of iterations, the computational time, and the final error, which was measured with a new added metric (mutual information).

II. Algorithm Analysis

A. Image Registration Framework

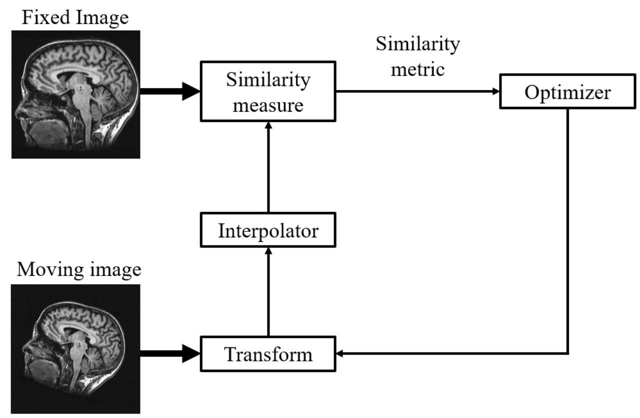

Fig. 1. Image Registration Framework

The figure II-B3 shows the stages and actors involved in an image registration framework. These are:

- Fixed image, which is the one used as base coordinate system.
- Moving image, which is the one that will be suffering transformations in order to be aligned with the fixed image.
- Transform, which includes all geometrical operations applied on the moving image in order to be aligned to the fixed image.
- Interpolator, which is used to estimate the pixel intensity values of the moving image given the position and the neighboring pixels.
- Similarity measure, which is typically a mathematical formula used to quantify how alike the images are at each step of the algorithm.
- Optimizer, which helps finding the best value for the evaluation metric depending to the parameters involved in the transformation.

Typically, the process in an image registration framework is the following: first, the moving image is transformed, then the interpolator is used for estimating the pixel intensity values for the moving image at that current step, and the similarity measure is used to determine the matching percentage between the two images. Then, the optimizer finds the best value for the evaluation metric based on the transformation parameters. This process is repeated in a loop until a convergence criterion is met (number of iterations, matching percentage, etc).

B. Similarity metrics

During this lab the following similatric metrics have been studied:

1) Squared differences: As it name suggests, this metric computes the square difference of intensities between the pixels of both images. For this metric, the lower the better, therefore this one implies a minimization problem. It is mathematically defined as follows:

$$SD = \frac{1}{N} \sum_{x=0}^{X} \sum_{y=0}^{Y} \left(A(x,y) - B(x,y)\right)^2$$

where N corresponds to the number of pixels in the images.

2) Normalized Cross Correlation:: This metric computes the correlation between pixels in all images divided by a scaling factor which is the squared root of the autocorrelation of each of the images. For this metric, high values denote high similarity between the images. Therefore, this implies a maximization problem. The metric is mathematically defined as follows:

$$R(A,B) = \frac{\sum_{x=0}^{X} \sum_{y=0}^{Y} (A(x,y) - \overline{I_A})(B(x,y) - \overline{I_B})}{\sqrt{\sum_{x=0}^{X} \sum_{y=0}^{Y} (A(x,y) - \overline{I_A})^2 \sum_{x=0}^{X} \sum_{y=0}^{Y} (B(x,y) - \overline{I_B})^2}}$$

In this equation, $A(x,y)$ and $B(x,y)$ denote the pixels intensities of images A, B, and $\overline{I_A}, \overline{I_B}$ denote the mean of pixel intensity values of images A and B.

3) Normalised Gradient Correlation: This metric is a bit similar to the cross-correlation, with the difference that in this case the gradient of the images is used, since according to [1] "the partial derivatives of image intensities form a better basis for achieving invariance to illumination change than using intensity values directly". For this metric, high values denote high similarity between the images, therefore, it also implies a maximization problem. It is mathematically defined as follows:

$$\Gamma(\mathbf{v}_1, \mathbf{v}_2) = \frac{\sum_{x,y} \left(\frac{\partial f_1(x,y)}{\partial x} \frac{\partial f_2(x,y)}{\partial x} + \frac{\partial f_1(x,y)}{\partial y} \frac{\partial f_2(x,y)}{\partial y} \right)}{\sqrt{\sum_{x,y} \left(\frac{\partial f_1(x,y)}{\partial x}^2 + \frac{\partial f_1(x,y)}{\partial y}^2 \right) \sum_{x,y} \left(\frac{\partial f_2(x,y)}{\partial x}^2 + \frac{\partial f_2(x,y)}{\partial y}^2 \right)}}$$

In this equation, the partial derivatives of f_1, f_2 denote the partial derivatives of the image intensities.

4) Mutual Information: As it name suggests, this metric measures the amount of information shared between the images (how much information one image contains about the other). It is mathematically defined as follows:

$$C(A,B) = H(A) + H(B) - H(A,B)$$

In this equation, $H(A), H(B)$ and $H(A,B)$ correspond to the entropy of images A, B and joint entropy of A,B, respectively. For each case, the entropy is computed using the following expression:

$$H(A) = -\sum p_i^b log p_i^b$$

C. Image Transformations

1) Rigid transformations: These types of transformations do not affect the shape or length of the images. It includes translations and rotations. It is mathematically represented as follows:

$$f_x(x,y) = x cos\phi + y sin\phi + t_x$$
$$f_y(x,y) = -x sin\phi + y cos\phi + t_y$$

where t_x and t_y denote translation in x and y angles, respectively, and ϕ represents the angle of rotation.

2) Affine transformations: These transformations modify the geometric structure of the image, but keeps the parallelism between lines. On top of translation and rotation, it includes other operations such as shearing and tearing. It is mathematically defined in the following way:

$$f_x(x,y) = a_x x + a_y y + t_x$$
$$f_y(x,y) = b_x x + b_y y + t_y$$

where t_x and t_y correspond to the parameters of translation in x and y axes, respectively, and a, b are the parameters for the other extra transformations.

D. Multiresolution Registration

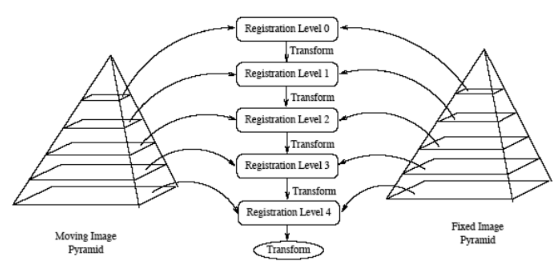

Fig. 2. Multiresolution registration framework [2]

As its name suggests, this process consists of performing the registration at different resolution levels (being this term a parameter of the algorithm) of the images in a pyramidal fashion. That being said, the bottom of the pyramid consists of the original image. At each level of the pyramid the image is shrink to a resolution that is the half of the previous one, and this is done up to the specified number of resolution levels. Starting from the top of the pyramid (smallest image) the registration is done and the output is used as the initialization for the next registration. This is repeated until the bottom of the pyramid which correspond to the original images. It is worth to highlight that when performing the registration procedure at each level, it is necessary to multiply the translation parameters by a factor of 2 when iterating downwards the pyramid, since the images were shrink to half of their resolution when the pyramid was built. This is done for every iteration except for the last one (when the final result is obtained).

III. Design and Implementation

For the design and implementation of the lab, the following work strategy was followed:

1) Implementation of new similarity metrics: for this part, the coding part consisted of translating the equations presented in the theoretical part to code in matlab.

A. Incorportation of affine transformations

in order to achieve this task, the affine transformation matrix was constructed and replaced in the code using the equation of the theoretical part. Therefore, it was the following matrix:

$$\begin{bmatrix} a_x & a_y & t_x \\ b_x & b_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

B. Multiresolution registration

The macroalgorithm for multiresolution registration is depicted in figure 3. It consists on two important loops: one for storing the resolution levels in a matrix (the pyramid), and a second one to perform the actual multiresolution registration by going downwards the built pyramid. At each iteration (except for the last one), the traslation parameters are multiplied by 2 and are used as initialization for the next registration.

C. Mutual Information

The macroalgorithm for MI computation is depicted in figure 4

D. Lab questions solutions

1) Question 1: What is the function of the scale vector?: Theoretically, the scale vector should be the learning rate of the optimizer. However, after doing experiments there was no noticeable role of this parameter.

2) Question 2:Where is the center of rotation of the transformation? : The center of rotation of the transformation is the center of the image coordinates.

IV. Experimental Results and Analysis

For this lab, Image registration was performed with four images ((a)brain1.png, (b)brain2.png, (c)brain3.png and (d)brain4.png) which can be shown in Figure 5. For ease of comparison, in all cases (b)brain2.png image was kept as fixed and images in a,c, and d were kept as moving images for registration. The images were registered using rigid (3 DOF) and affine (6 DOF) transformations along with metrics such as squared distance (SD), Normalized Cross-Correlation (CC), and Gradient Normalized Cross-Correlation (GCC). The

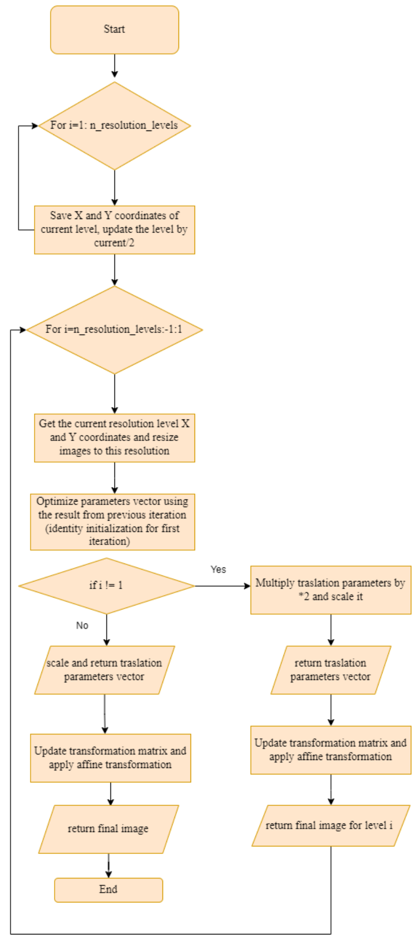

Fig. 3. Multiresolution registration macro-algorithm

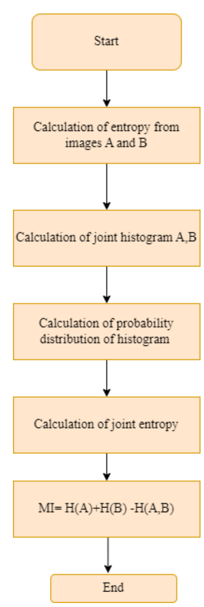

Fig. 4. Mutual Information calculation macroalgorithm.

TABLE I

RESULTS OBTAINED FOR RIGID TRANSFORMATION.

Fixed Image	Moving Image	Resolution	Metric	Mutual Information	Computational Time (s)
brain2.png	brain1.png	Single	SD	5,9876	24,314
	brain3.png			0,6181	13,277
	brain4.png			0,3321	32,385
brain2.png	brain1.png	Single	CC	6,0331	32,484
	brain3.png			0,7173	31,775
	brain4.png			0,6074	8,021
brain2.png	brain1.png	Single	GCC	0,6637	22,848
	brain3.png			0,6151	14,794
	brain4.png			0,5672	20,699
brain2.png	brain1.png	Multi	SD	5,6921	29,064
	brain3.png			0,6181	26,596
	brain4.png			0,3341	51,354
brain2.png	brain1.png	Multi	CC	5,1187	28,541
	brain3.png			0,6186	26,989
	brain4.png			0,299	36,688
brain2.png	brain1.png	Multi	GCC	4,6643	50,284
	brain3.png			0,6105	41,649
	brain4.png			0,5403	67,647

TABLE II

RESULTS OBTAINED FOR AFFINE TRANSFORMATION.

Fixed Image	Moving Image	Resolution	Metric	Mutual Information	Computational Time (s)
brain2.png	brain1.png	Single	SD	0,7	37,193
	brain3.png			0,6601	39,146
	brain4.png			0,542	51,493
brain2.png	brain1.png	Single	CC	1,3966	49,517
	brain3.png			0,6519	43,267
	brain4.png			0,233	20,253
brain2.png	brain1.png	Single	GCC	0,6753	50,108
	brain3.png			0,6149	58,195
	brain4.png			0,5566	60,679
brain2.png	brain1.png	Multi	SD	5,969	88,384
	brain3.png			3,7645	72,829
	brain4.png			0,4063	116,729
brain2.png	brain1.png	Multi	CC	6,0351	95,872
	brain3.png			3,7653	65,997
	brain4.png			0,2483	125,923
brain2.png	brain1.png	Multi	GCC	6,0874	110,569
	brain3.png			3,8896	98,636
	brain4.png			0,6658	138,556

performance using single and multi-resolution image registration was also compared. In all the experiments, some hyper-parameters are kept constant to compare the results with each other. For example, in the optimizer maximum number of iterations was 1000, the termination tolerance on the function was $10^{-}9$, the termination tolerance on parameters was $10^{-}9$ and a maximum number of function evaluations was 1000 × length(parameters). No experiments were done to tune the hyperparameters. In the following sections, the performance in each of the cases discussed above will be analyzed and explained.

A. *Rigid Transformation with Single Resolution*

For the first case, resolution parameter was kept 1 and both original images were used for registration choosing the transformation rigid and varying the metrics. The initialization parameters were chosen as x = [0 0 0] and scale = [0.1 0.1 1]. From the image chart 8 we can see, Rigid transformation with single resolution works very good for image1 and image2 but due to have only 3 DOF, it performs bad for image3 with image2. Also, due to the changes in illumination in the background, image2 and image4 registration was poor. Both SD and CC metric works well when the images are only translated and rotated but if the images are also scaled or sheared, they perform not very good. Moreover, Images with GCC metric does not perform satisfactorily. This might cause because of poor initialization or poor hyper parameter combination. Table I also shows the mutual information scores

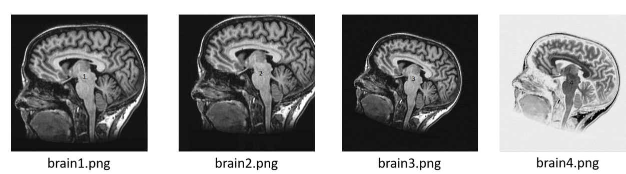

Fig. 5. Images Used for Image Registraion: fixed: (b) and moving (a),(c),and (d).

and computational time of each of the registration operation with this case where computational time shows that GCC metric normally takes the shortest time and CC the longest for multiple resolution cases to reach the convergence.

B. Rigid Transformation with Multi Resolution

For the second case, resolution parameter was kept 5 and pyramid of multiple resolution images were used while choosing the transformation rigid and varying the metrics. The initialization parameters were chosen as x = [0 0 0] and scale = [0.1 0.1 1]. As the registration started with lower resolution, after each step the translation parameter was doubled by x= x.*[2 2 1 1 1 1] this operation. From the image chart 9 we can see, Rigid transformation with multiple resolution works very good for image1 and image2 with all the metrics as multiple resolution solves the problem of bad initialization with a trade off of computational time increase. Due to less degree of freedom and change in illumination background, still image3 and image4 does not perform well with image2. Table I also shows the mutual information scores and computational time of each of the registration operation with this case computational time shows that SD metric normally takes the shortest time and GCC the longest for single resolution cases to reach the convergence.

C. Affine Transformation with Single Resolution

For the third case, resolution parameter was kept 1 and both original images were used for registration choosing the transformation affine and varying the metrics. The initialization parameters were chosen as x = [0 0 0 1 1 0] and scale = [0.1 1 0.1 1 1 1]. From the image chart 10 we can see, Affine transformation with single resolution works very bad for all cases as affine has more degree of freedom so the initialization of the parameters plays a key role. Table I also shows the mutual information scores and computational time of each of the registration operation with this case computational time shows that SD metric normally takes the shortest time and GCC the longest for single resolution cases to reach the convergence.

D. Affine Transformation with Multi Resolution

For the last case, resolution parameter was kept 5 and pyramid of multiple resolution images were used while choosing the transformation affine and varying the metrics. The initialization parameters were chosen as x = [0 0 0 1 1 0] and scale = [0.1 1 0.1 1 1 1]. As the registration started with lower resolution, after each step the translation parameter was doubled by x= x.*[2 2 1 1 1 1] this operation. From the image chart 8 we can see, Affine transformation with multiple resolution works very good for image1, image3 with image2 with all the metrics as multiple resolution solves the problem of bad initialization with a trade off of computational time increase. Due to high intensity difference in illumination background, still image4 does not perform well with image2. Table I also shows the mutual information scores and computational time of each of the registration operation with this case where computational time shows that SD metric normally takes the shortest time and GCC the longest for multiple resolution cases to reach the convergence.

V. Visual Analysis

For Visual Analysis, two cases (best case with image 1 and image 2 also worst case with image 4 and image 2) were picked and for them bar charts were plotted to show the comparison with respect to Mutual Information and Computaional Time. Figure 6 and 7 demonstrates the comparison as mentioned. Moreover, for these two cases (single resoution) iteration vs function optimization value were also plotted for all the metrics which can be found in the appendix from Figure 12 to Figure 23.

VI. Project Management and Details

For this experiment, one problem arised due to initialization of the affine transformation matrix and then solved it by starting with an identity matrix. Later, the doubling the translation parameter in case of multiple resolution was also solved. Other than that, no problems were faced and the workload has been divided equally between the partners both for coding and writing part.

VII. Conclusion

In the first lab, Intensity based image registration was performed with four images. Normally, Rigid transform works well when there is only need of performing translation and rotation due to having less degree of freedom. But Affine works better as it has more degree of freedom and can perform more image transformations like scaling and shearing. But also, more degree of freedom needs proper initialization of the parameters to reach convergence. Poor initialization problem can be solved by using pyramid like multi resolution-based image registration with a trade off of computational time increase. Hyper parameters can also be tuned using grid search for better optimization. Still multi-resolution fails when there is high intensity difference between the images for which we need more advanced image registration process. The take way of this lab will be no transformation or metric can be superior to others where as choice of transformation or metric largely depends on the images to be registered.

References

[1] R. O'Callaghan and T. Haga, "Robust change-detection by normalised gradient-correlation," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[2] R. Martí, "Mira: Week 1: Intensity based image registration," in *Lecture Slides*, 2022, p. 7.

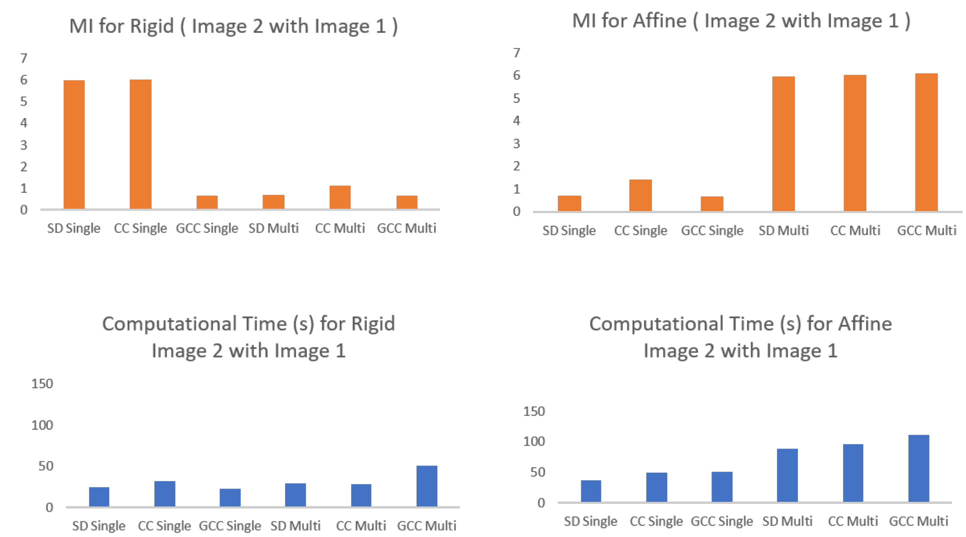

Fig. 6. Case 1: Image 2 (fixed) with Image 1 (moving).

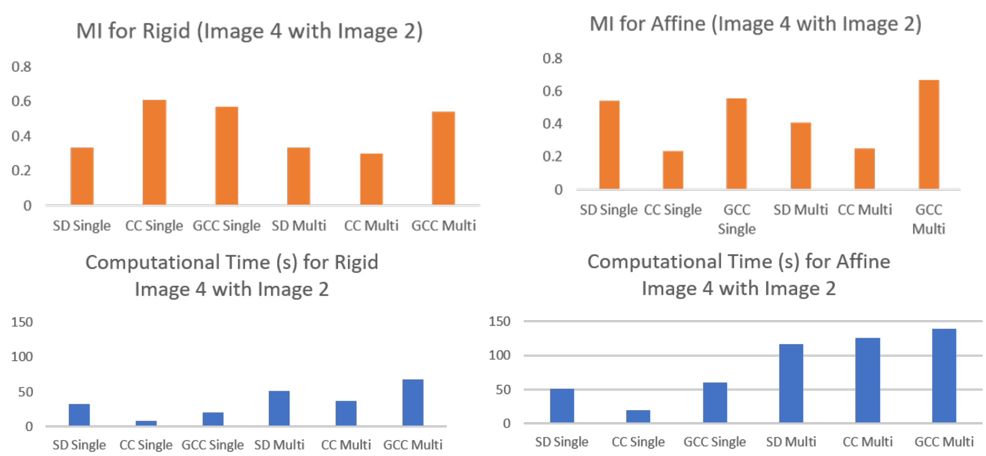

Fig. 7. Case 1: Image 2 (fixed) with Image 1 (moving).

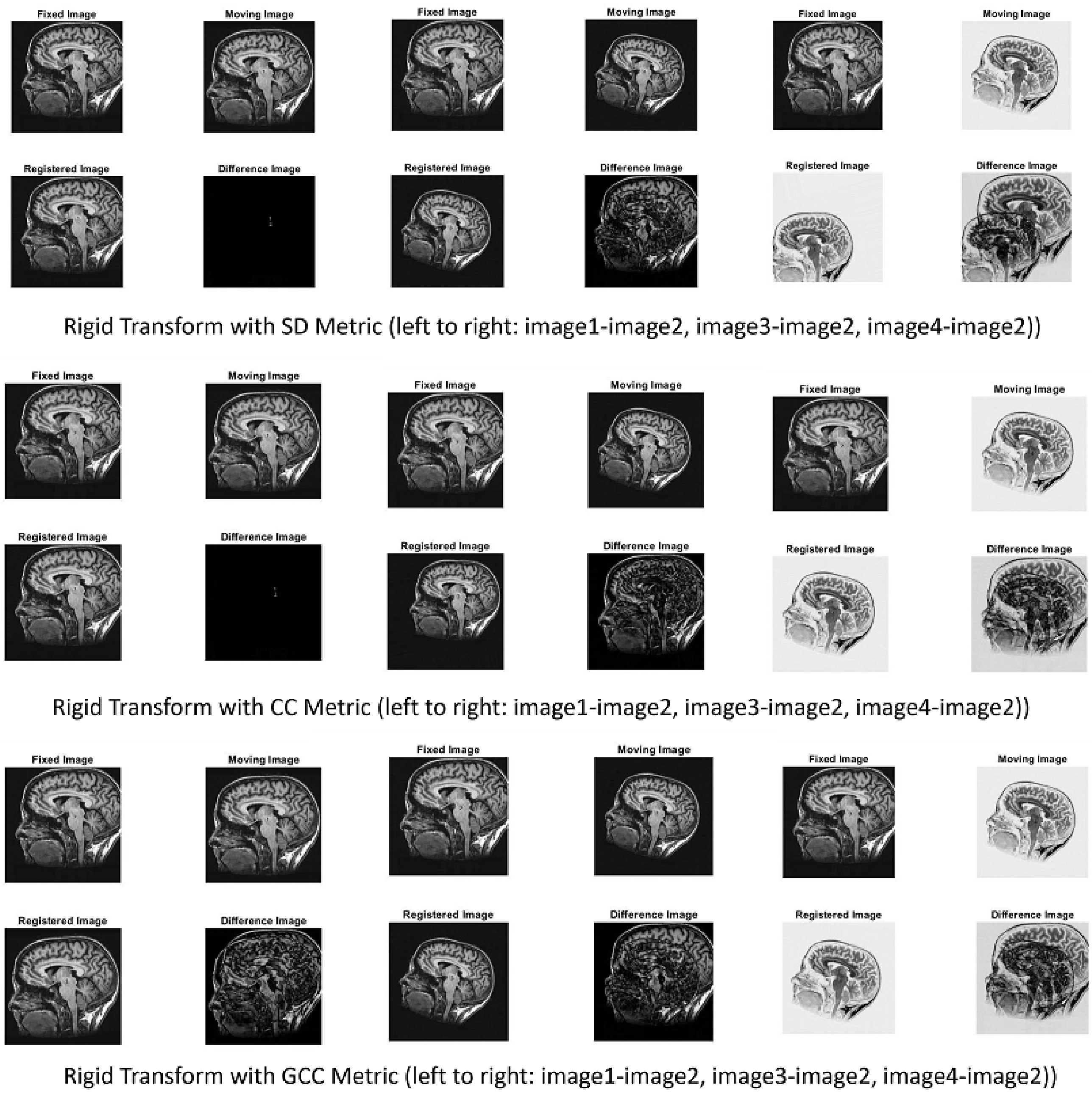

Fig. 8. Rigid Transforms with Single Resolution.

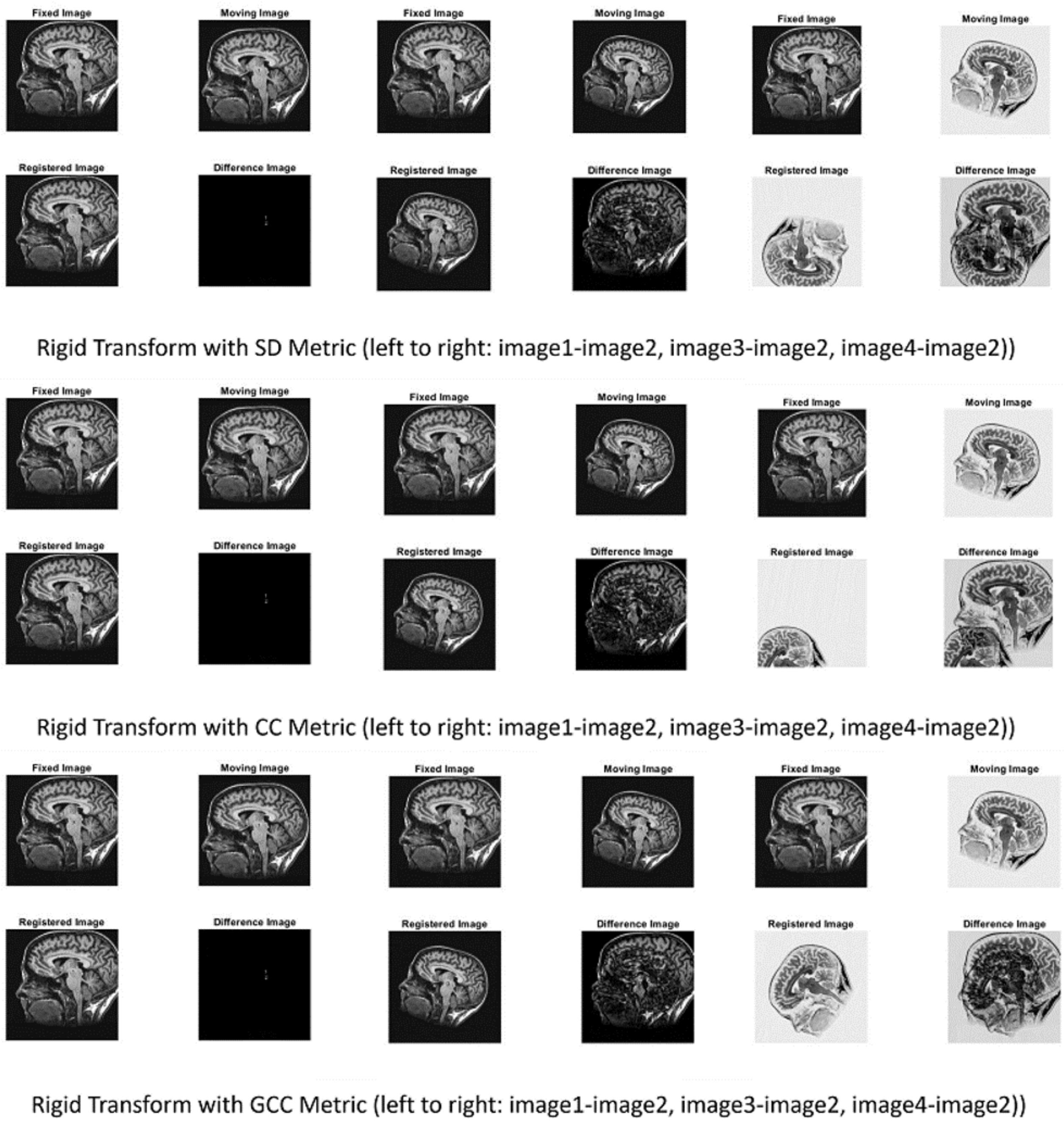

Fig. 9. Rigid Transforms with Multi-Resolution.

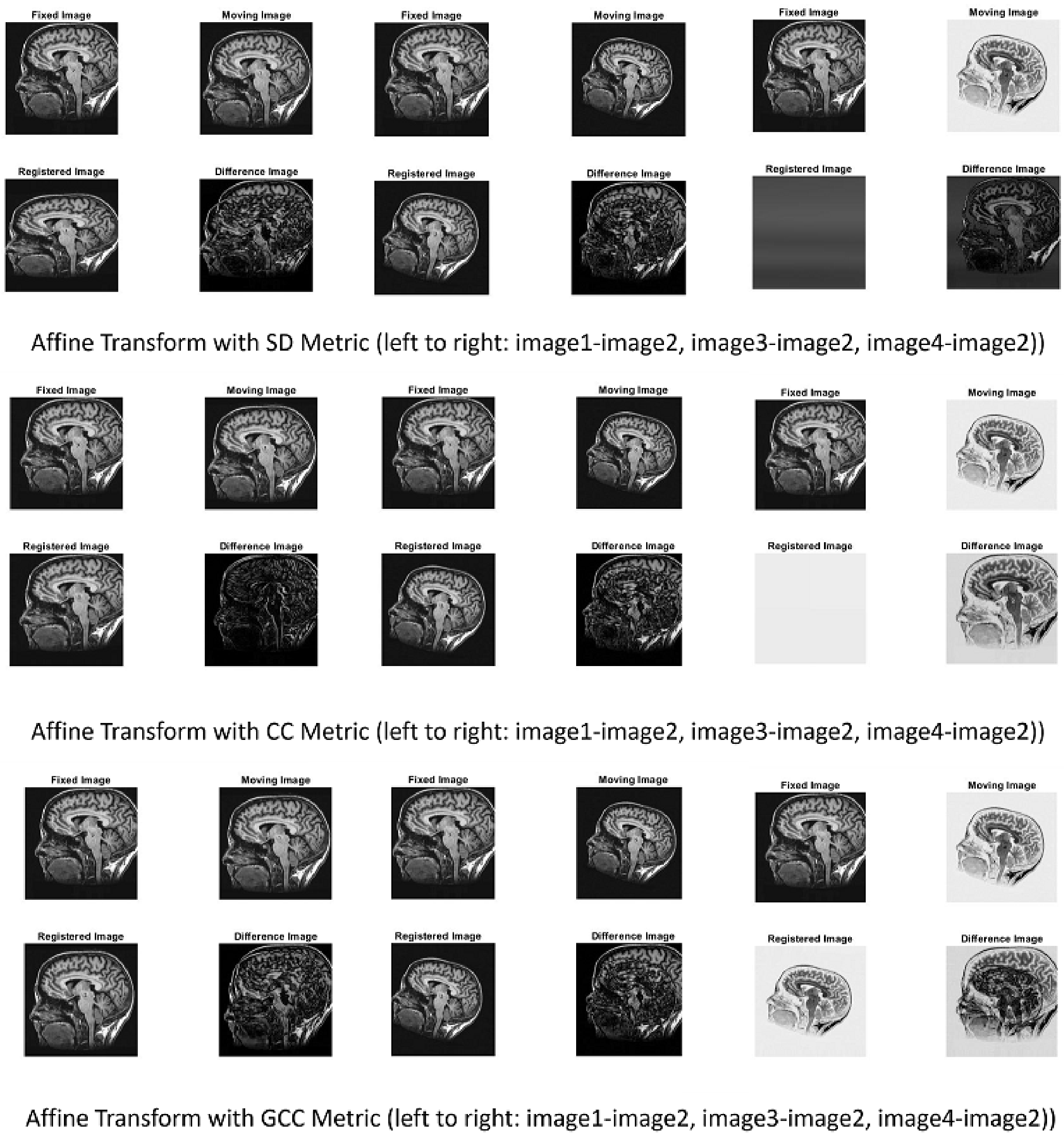

Fig. 10. Affine Transforms with Single Resolution.

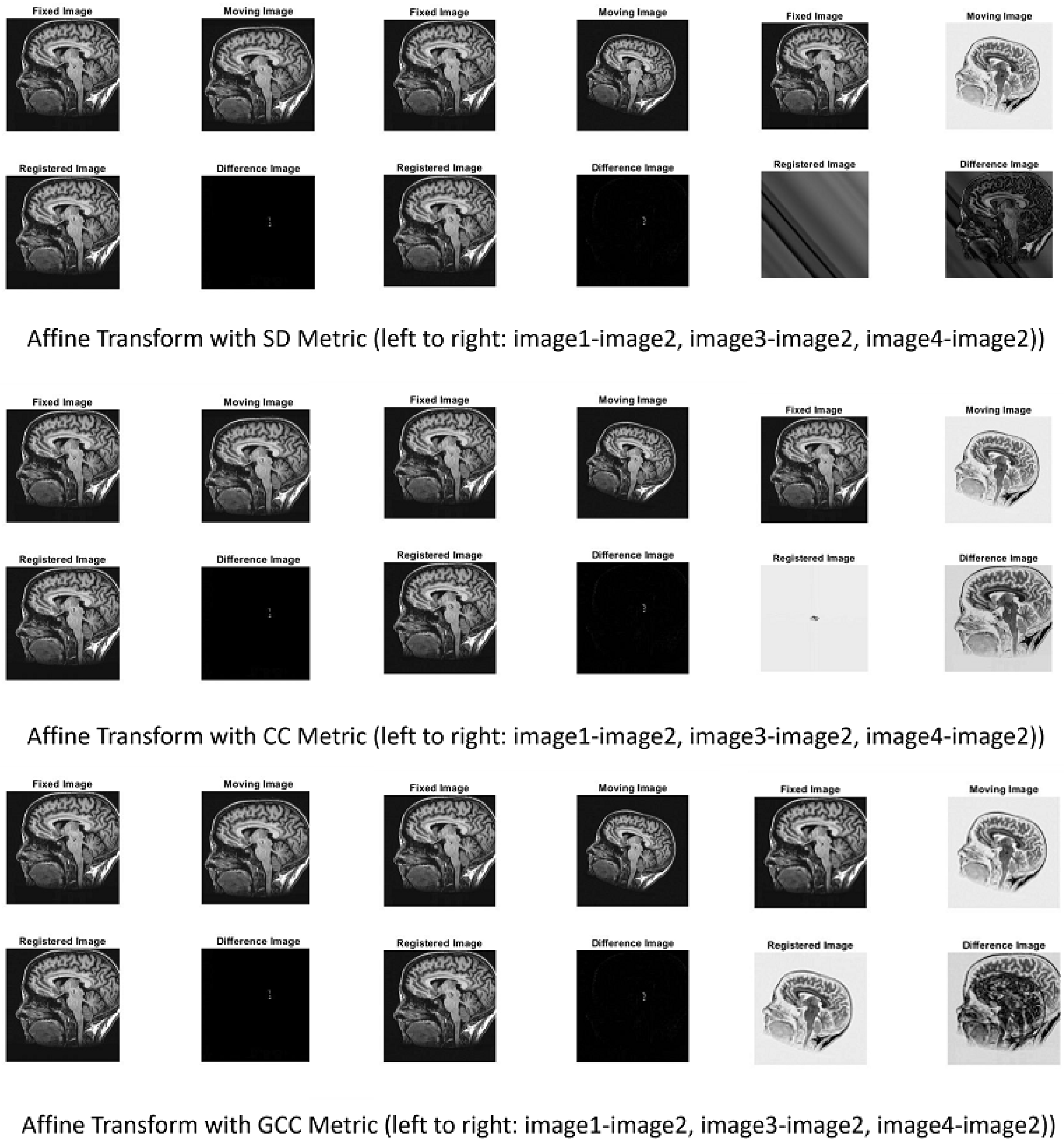

Fig. 11. Affine Transforms with Multi-Resolution.

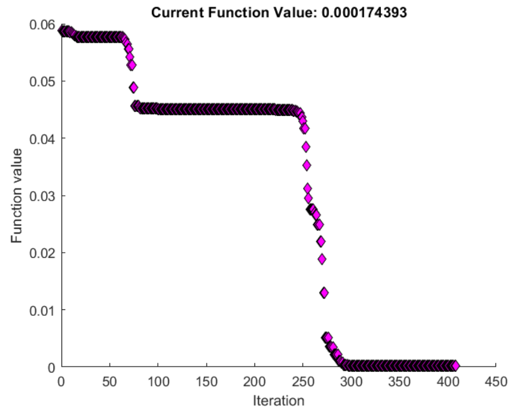

Fig. 12. Metric against iterations for case 1: image 1 moving, image 2 fixed, rigid registration, single resolution, SD metric

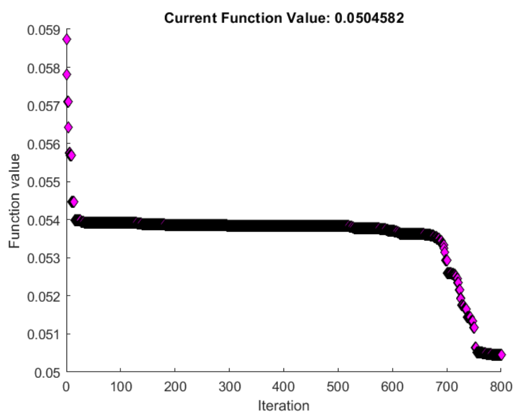

Fig. 13. Metric against iterations for case 2: image 1 moving, image 2 fixed, affine transformation, single resolution, SD metric

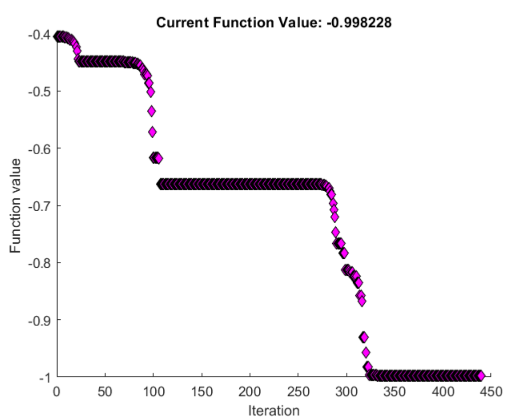

Fig. 14. Metric against iterations for case 3: image 1 moving, image 2 fixed, rigid transformation, single resolution, CC metric

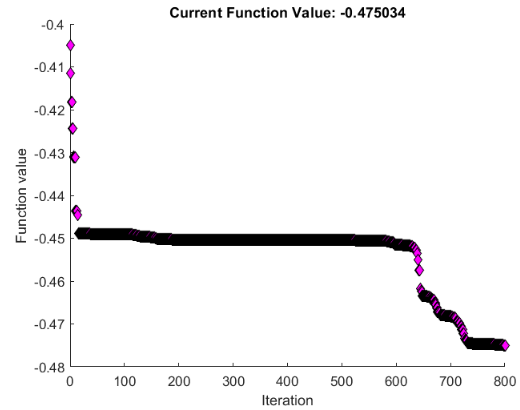

Fig. 15. Metric against iterations for case 4: image 1 moving, image 2 fixed, affine transformation, single resolution, CC metric

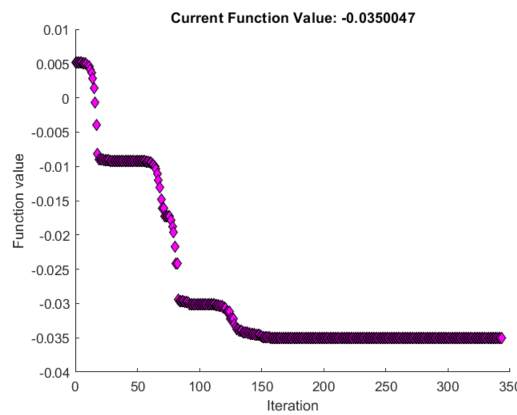

Fig. 16. Metric against iterations for case 5: image 1 moving, image 2 fixed, rigid transformation, single resolution, GCC metric

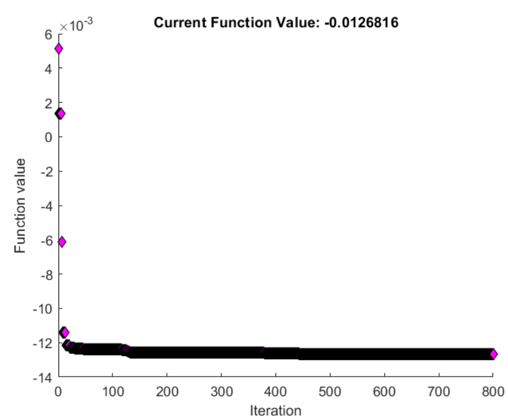

Fig. 17. Metric against iterations for case 6: image 1 moving, image 2 fixed, affine transformation, single resolution, GCC metric

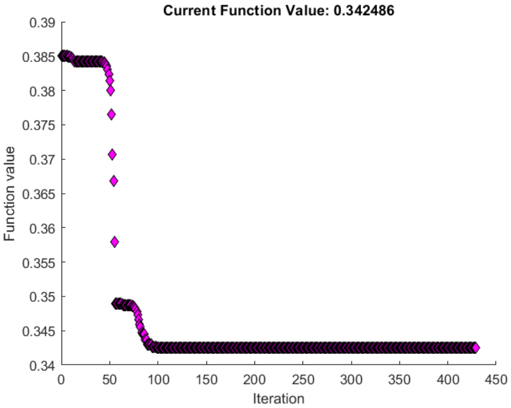

Fig. 18. Metric against iterations for case 7: image 4 moving, image 2 fixed, rigid transformation, single resolution, SD metric

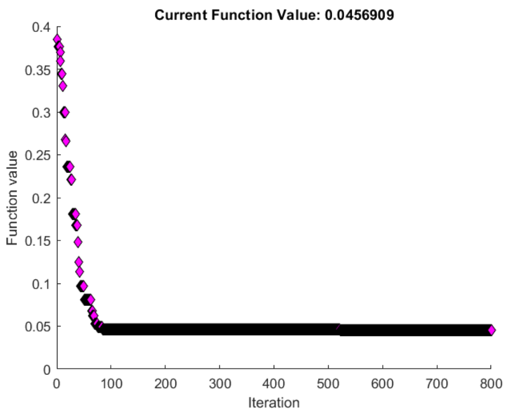

Fig. 19. Metric against iterations for case 8: image 4 moving, image 2 fixed, affine transformation, single resolution, SD metric

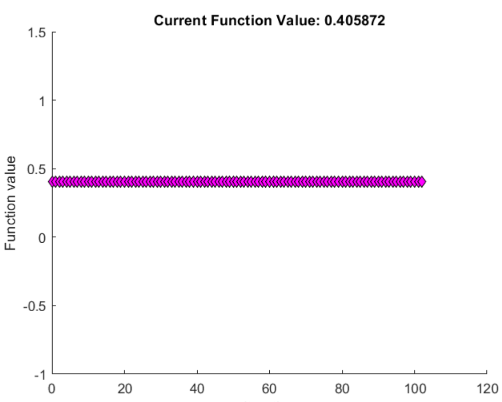

Fig. 20. Metric against iterations for case 9: image 4 moving, image 2 fixed, rigid transformation, single resolution, CC metric

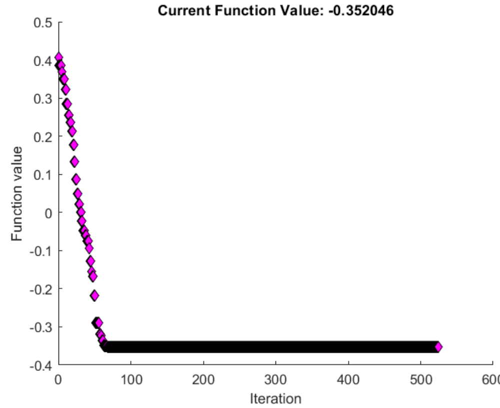

Fig. 21. Metric against iterations for case 10: image 4 moving, image 2 fixed, affine transformation, single resolution, CC metric

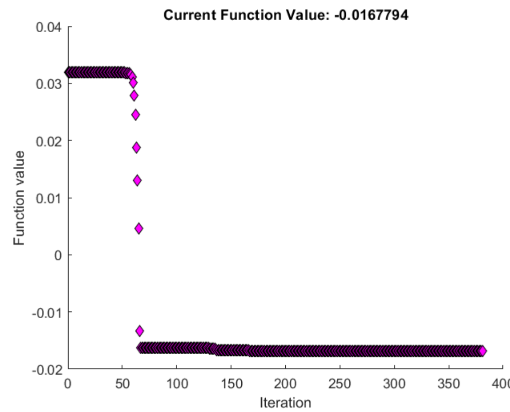

Fig. 22. Metric against iterations for case 11: image 4 moving, image 2 fixed, rigid transformation , single resolution, GCC metric

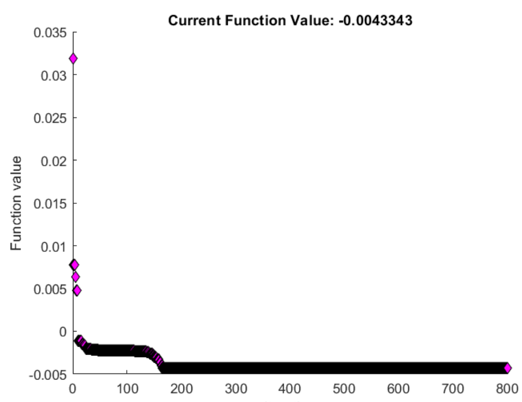

Fig. 23. Metric against iterations for case 12: image 4 moving, image 2 fixed, affine transformation, single resolution, GCC metric