Universitat de Girona

Medical Imaging and Applications

Medical Image Registration and Applications Course

Final Project 18/19
TRE-Driven Inhale-Exhale Intra-patient Registration Using Dir-Lab Dataset

Authors:
Basel Alyafi
Md. Kamrul Hasan
Fakrul Islam Tushar

Supervisor:
Dr. Robert Marti
Dr. Rafael Garcia

January 28, 2019

Contents

1 Introduction and Problem Definition

Image registration is an essential process to draw correspondence between images [1]. It is usually done by aligning two images, or more, conventionally named as fixed and moving. The fixed image usually lies in the preferred space to which we want to move the moving image(s) [2]. Intensity-based image registration has multiple types, but most importantly are the rigid and non-rigid ones. Rigid transformation is usually used for mapping images that differ in regular and general patterns, like scale, rotation, translation and shearing/tearing. In this case, the non-local transformation is applied on all pixels similarly. On the other hand, non-rigid registration applies local transformation functions. In this case, there is no one general pattern that can be detected while deformations happen differently in different regions in the image. The former type is usually used as an initialization for the latter one which has a higher computational complexity. This mechanism is offered by `Elastix` [3] [4].

The aim of this project is to minimize the difference between the calculated features' locations and the physical ones. That is done by registering the extreme exhale lung CT image (moving) to the extreme inhale (fixed) of the same patient applied on four different patients taken from `dir-lab COPDgene` dataset (COPD1, COPD2, COPD3, COPD4) website. Each CT volume comes with a 300-feature pixel indices text file. The idea is to find the exhale pixel indices given the inhale ones by applying the transformation got from registering the volumes. Performance evaluation is done by calculating the average (and standard deviation of) 3-dimensional Euclidean distances (sometimes referred to as Target Registration Error or TRE) between corresponding features' locations, for the mathematical formula, check section 3.3.

2 Environment

`Elastix` was used for registering 3D CT thoracic volumes, while `Transformix` was used to transform the points (pixel indices) from fixed (extreme inhale) to moving space (extreme exhale). Matlab was also used especially for reading files and calculating the objective evaluation function (TRE_stats) shown in Algorithm 1. For information about the machine used, check section 4.2.

2.1 Dir-Lab Dataset

In this coursework, 4 COPDgene volumes (1, 2, 3, and 4) were taken from dir-lab website (for the link, check section 1) and registered in an intra-subject manner (exhale to inhale of the same patient). Table 1 shows the dimensions, voxel spacing and number of features for each case. Displacements were calculated by using eqs(2, 3). Observers values were taken from the website and were computed originally on a different higher number of features but used here as a reference. Images were accompanied with corresponding features locations

Label	**Dimensions**	**Voxel Spacing** (mm)	**Displacement** $\mu(\sigma)$	**Observers (mm)**
COPD1	512 ×512 × 121	0.625 ×0.625 × 2.5	26.33(11.43)	0.65(.73)
COPD2	512 ×512 × 102	0.645 ×0.645 × 2.5	21.78(6.471)	1.06(1.51)
COPD3	512 ×512 × 126	0.652 ×0.652 × 2.5	12.639(6.395)	0.58(0.87)
COPD4	512 ×512 × 126	0.59 ×0.59 × 2.5	29.58(12.94)	0.71(0.96)

Table 1: Dataset description

files, each file consists of 300 lines (each with x, y, z coordinates of one feature), those values represent pixel indices. Those files were passed to `Transformix` to transform the points using the registration between corresponding volumes. Reading files were accomplished by using `Matlab textscan` function.

3 Dataset Registration and Tuning

In this section, the whole process from reading raw images until reaching the last stage of tuning is explained with pictures.

3.1 Data Rehabilitation

Raw images (binary files) were loaded by using `ITK-SNAP`. Information from Table 1 were input and one Superior-Inferior rotation was done on all images to ensure correct positions of all features. After that, modified raw images were saved as `nifti` images and saved for next processes.

3.2 Initialization

Firstly, registration was done mainly in Elastix [3] [4]. The process can be described as follows:

1. Rigid registration using parameter file shown in Table 2.
2. Non-rigid registration (BSpline) applied on the result of the former step with initial parameters shown in Table 3.

To go through the important parameters briefly, multi resolution registration with 6 levels were examined to be working well in most cases. The metric was under test and was changed later on, random image sampler with smoothing pyramids (no resizing) ensures good balance between computational complexity and performance. Grid spacing for the non-rigid registration was under test and was changed later on. Optimizer parameters estimation was selected to be displacement distribution which lets `Elastix` determines the values of optimization parameters adaptively. Formats were selected to include compression to minimize files sizes. An important point which should be understood clearly is how we selected the fixed and moving images, see Figure 1.

//*********Components
(Registration "MultiResolutionRegistration")
(Metric "AdvancedNormalizedCorrelation")
(ImageSampler "RandomCoordinate")
(Interpolator "LinearInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(Transform "AffineTransform")
(Optimizer "AdaptiveStochasticGradientDescent")
// ********* Pyramids
(NumberOfResolutions 5)
(FixedImagePyramid "FixedSmoothingImagePyramid")
(MovingImagePyramid "MovingSmoothingImagePyramid")
// ********* Transform
(AutomaticScalesEstimation "true")
(HowToCombineTransforms "Compose")
(AutomaticTransformInitialization "true")
// ********* Optimizer
(ASGDParameterEstimationMethod "DisplacementDistribution")
// ********* Interpolator
(FinalBSplineInterpolationOrder 3)
// ********* Data types and formats
(FixedInternalImagePixelType "float")

(MovingInternalImagePixelType "float")
(ResultImageFormat "nii.gz")
(ResultImagePixelType "float")
(WriteResultImage "false")
// ********* **Image Sampler** (MaximumNumberOfSamplingAttempts 20)
(NewSamplesEveryIteration "true")

Table 2: initial rigid registration parameters

//*********Components (Registration "MultiResolutionRegistration")
(Metric "AdvancedNormalizedCorrelation")
(ImageSampler "RandomCoordinate")
(Interpolator "LinearInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(Transform "BSplineTransform")
(Optimizer "AdaptiveStochasticGradientDescent")
// ********* **Pyramids** (NumberOfResolutions 5)
(FixedImagePyramid "FixedSmoothingImagePyramid")
(MovingImagePyramid "MovingSmoothingImagePyramid")
// ********* **Transform** (AutomaticScalesEstimation "true")
(HowToCombineTransforms "Compose")
(AutomaticTransformInitialization "true")
(FinalGridSpacingInVoxels 8.0 8.0 8.0)
// ********* **Optimizer**
(ASGDParameterEstimationMethod "DisplacementDistribution")
// ********* **Interpolator**
(FinalBSplineInterpolationOrder 3)

// ********* **Data types and formats**
(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
(ResultImageFormat "nii.gz")
(ResultImagePixelType "float")
// ********* **Image Sampler**
(MaximumNumberOfSamplingAttempts 20)
(NewSamplesEveryIteration "true")

Table 3: initial non-rigid registration parameters

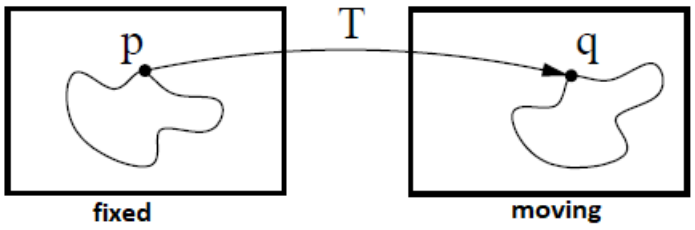


Figure 1: How Elastix computes the transformation, the direction is from the fixed image towards the moving one. [5]

According to this figure, the fixed image should be the inhale one, the moving image should be the exhale one. That is to be able to compute the locations of the landmarks in the exhale image given the fixed ones (and both images!).

3.3 Objective Evaluation

The objective was to minimize the TRE value between the registered and original landmarks for each volume. TRE as shown in Figure 2 is important to be minimized in order to have good registered landmarks. In Figure 2, x_2 with a black crosshair is the actual location given by specialists (from dir-lab website), while $\hat{x}_2$ with a green crosshair is the registered landmark (estimated locations). How the average and the standard deviation of TRE are computed is as eq(2) and eq(3); where S_x, S_y, S_z are the voxel spacing values over x, y, z, respectively.

$$d_i = \sqrt{[(x_i - \hat{x}_i) \times S_x]^2 + [(y_i - \hat{y}_i) \times S_y]^2 + [(z_i - \hat{z}_i) \times S_z]^2} \tag{1}$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} d_i \tag{2}$$

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} |d_i - \mu|^2} \tag{3}$$

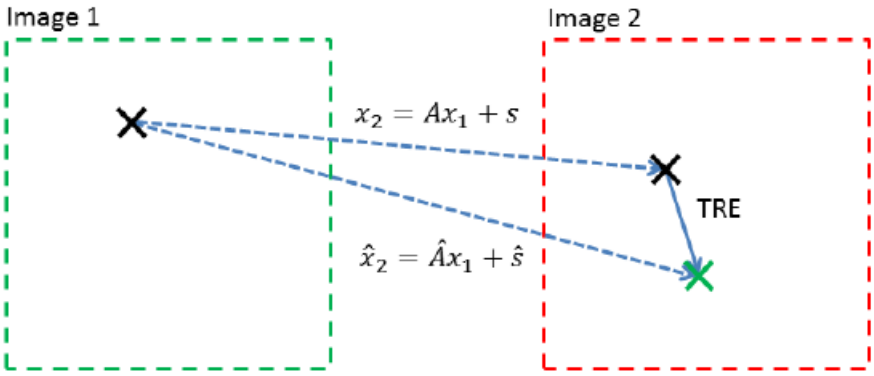


Figure 2: TRE meaning [6], image 1 is the fixed one, image 2 is moving, the black crosshair is the groundtruth, the green one is the computed.

3.4 Registration Process

The main algorithm for doing the registration is shown in Algorithm 1. At the beginning, landmarks indices are read from text files. Then, the images were registered by using `Elastix` and parameter files shown in tables 2 and 3. The estimated landmarks indices are computed using `Transformix` and the transform parameters resulted from previous registration. Fianlly, the mean and standard deviation are computed from the estimated indices and ground truth (the original moving indices) using eq(2, 3).

Algorithm 1 The main registration algorithm (training phase)

1: fix_landmarks = $read(inhale\_landmark)$
2: mov_landmarks = $read(exhale\_landmark)$
3: reg_img, transform_param =

$$\texttt{Elastix}(fix = inhale\_img,\ mov = exhale\_img,\ init = rigid\_param,\ param = nonrigid\_param)$$

4: $estimated\_landmarks = \texttt{Transformix}(fix\_landmarks,\ transform\_param)$
5: Compute the average 3D distance and standard deviation

$$\mu, \sigma = TRE\_stats(mov\_landmarks,\ estimated\_landmarks)$$

3.5 Tuning Process

Tuning was done by repeating Algorithm 1 several tens of times after doing changes to the parameter files (rigid and non-rigid) and measuring how close the results are to the observers values shown in Table 1. This process included 10 effective changes in parameter files during an exhaustive investigation and exploration. Looking for perfect parameters were similar to finding a needle in a haystack!. A large number of combinations between different parameters were examined and the stages were applied on both parameter files, unless stated otherwise, as follows:

1. `(initial)` use the initial parameter files.

2. `(PyramResol)` increase the number of resolutions of the pyramids in both files.

3. `(z Grid)` decrease the BSpline grid spacing on Z axis to 4 due to the fact that it has lesser resolution (higher voxel spacing) than on other axis.

4. `(z PyramSmooth)` decrease the smoothing level when creating the pyramids on Z axis due to the same previous reason. The schedule used was
`(ImagePyramidSchedule 12 12 5  10 10 4  8 8 3  4 4 2  2 2 1  1 1 1)` while the default one was
`(ImagePyramidSchedule 32 16 8 4 2 1)` .

5. (`ROI`) the use region of interest was an important step that boosted the performance significantly. The idea is to randomly sample pixels out of a predefined cube (in our case of size $150 \times 150 \times 35$) instead of sampling from the whole image. It is kind of local optimization.

6. (`pyramSmooth`) changing the pyramids smoothing schedule to
`(ImagePyramidSchedule 14 14 3  10 10 2  8 8 2  4 4 1  2 2 1  1 1 1)`.

7. (`opt`) optimizer parameters were changed in this step, the learnin-rate related parameter (`SP_A`) was changed from 20 to 40 then 50. It is recommended in [5] to use 10% of the number of iterations (500 in this case).

8. (`MI bins`) the metric in the final parameter file was changed to MutualInformation with 64 histogram bins.

9. (`samples`) the default number of samples taken by the image sampler is 5000. It was worked out that 25000 was just better.

10. (`pyramSmooth`) for the last time, the smoothing level was tuned to cope with the heterogeneity of voxel spacing on different axis.

    Performance evaluation was done after each single change and numbers were written down for later comaprison. After each change, all four results were compared with previous ones to ensure robustness. For checking the numerical performance, check section 4

To check the final changes in parameter files, see Appendix A and Appendix B.

4 Results

To see the effect of each step in section 3.5, Figures 3, 4, 5, and 6 show the accumulative effect of tuning stages on images COPDgene 1 and 2, 3, and 4 respectively. That is, the result at any point include all previous unrelated changes. It can be seen from all those figures that `ROI, MI bins`, and `samples` played a great role to reach the final result. For each of the figures, the last bar on the right represents the final precision reached. By comparing Figures 3, 4, 5, and 6, the performance order is as follows:

1. COPD1 with 1.45(1.7712).

2. COPD3 with 1.781(2.093).

3. COPD4 with 2.6508(2.9983).

4. COPD3 with 5.1039(6.2511).

To draw a simple analysis, by comparing image dimensions in Table 1, it can be seen that image 2 has less pixels on Z axis which might make it more sensitive to smoothing (pyramids) and grid scaling (BSpline). On the other hand, all other three images have similar number of pixels. Additionally by looking at observers values (and best results on dir-lab website), COPDgene 2 has the lowest precision in all cases (highest mean and standard deviation).

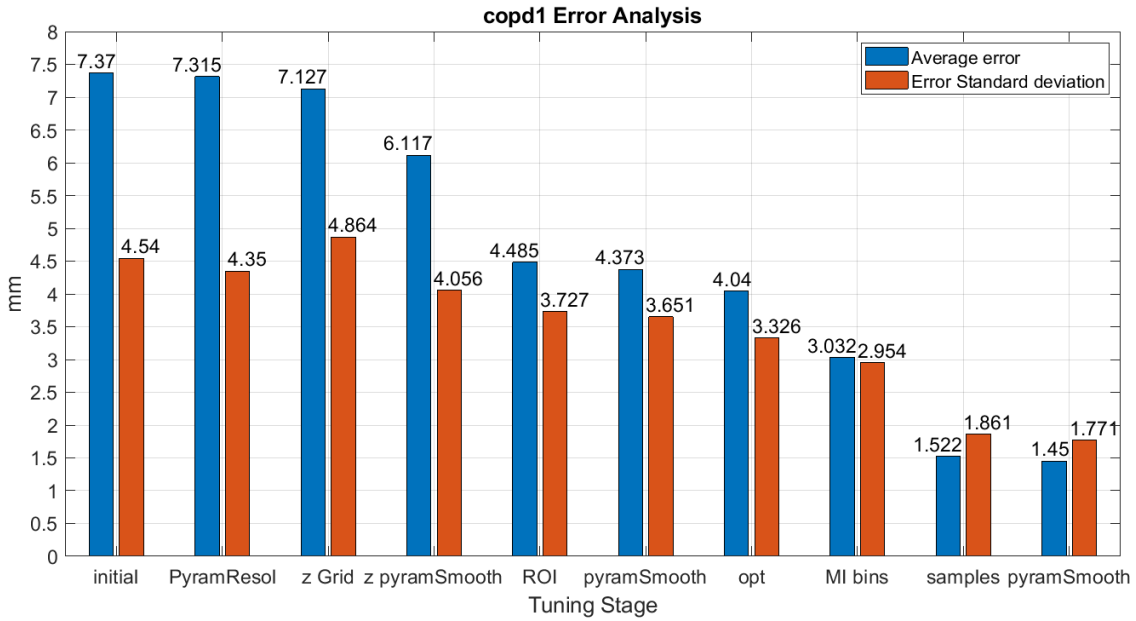


Figure 3: COPDgene 1 progress

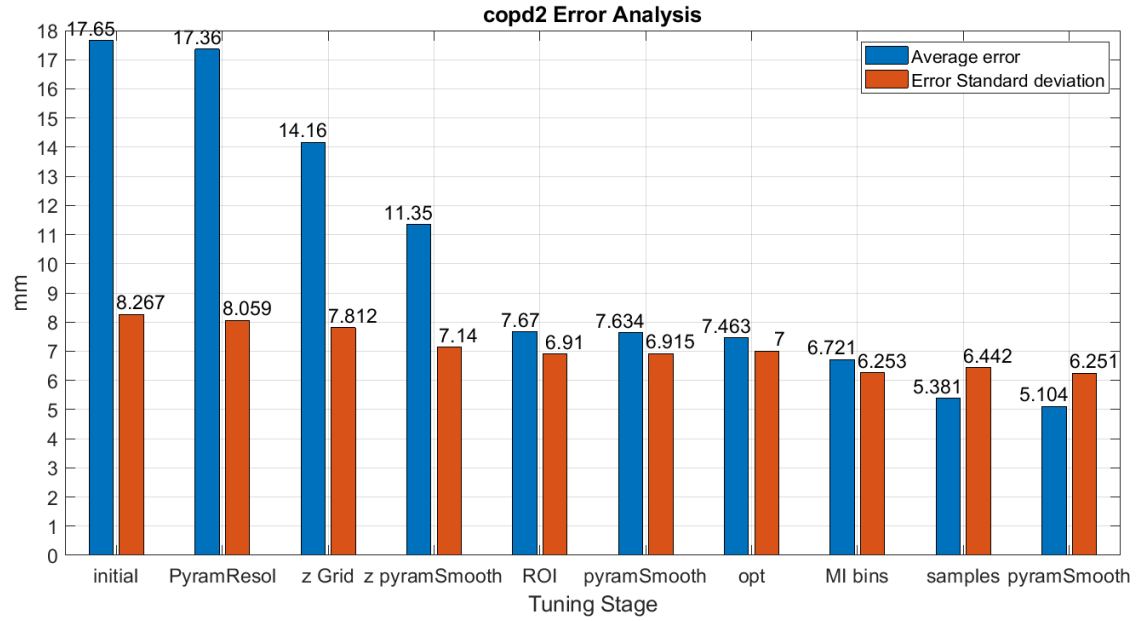


Figure 4: COPDgene2 progress

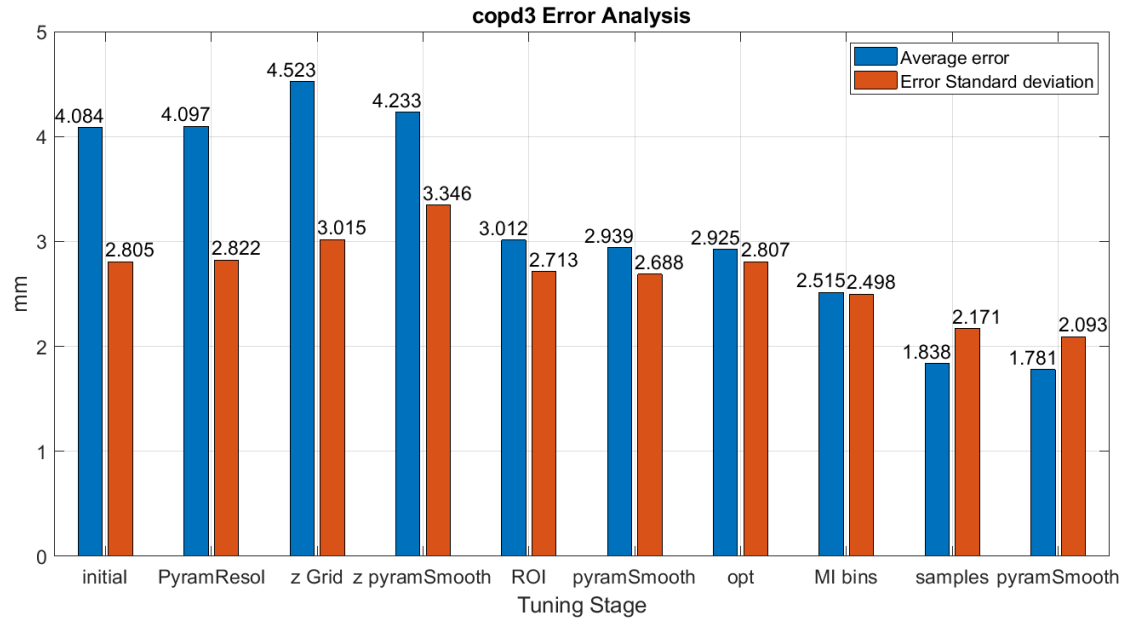


Figure 5: COPDgene3 progress

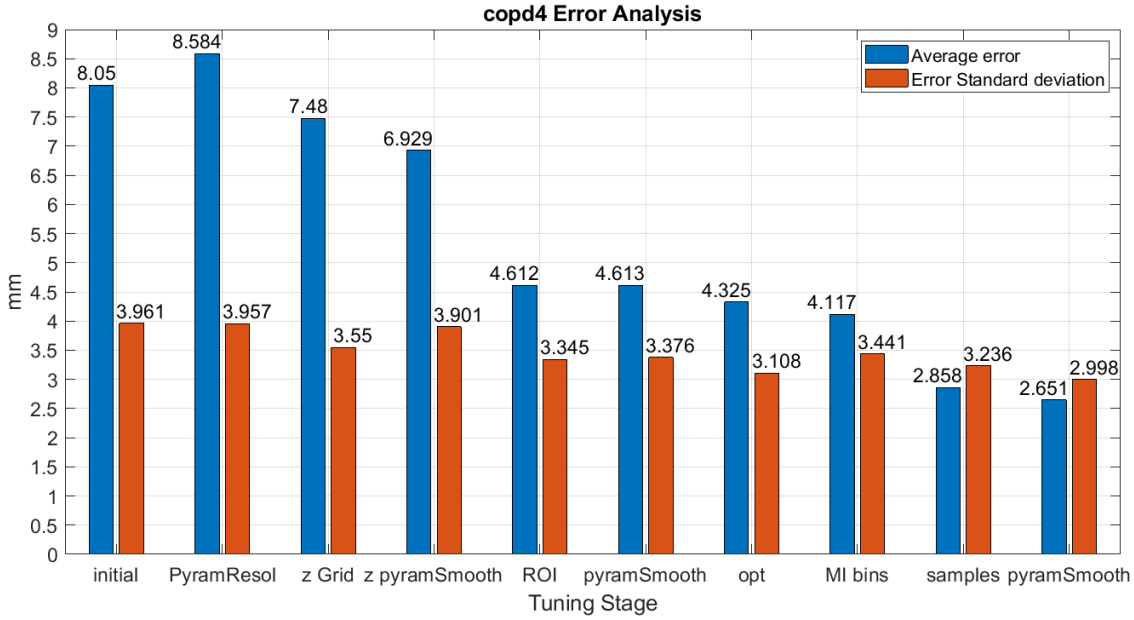
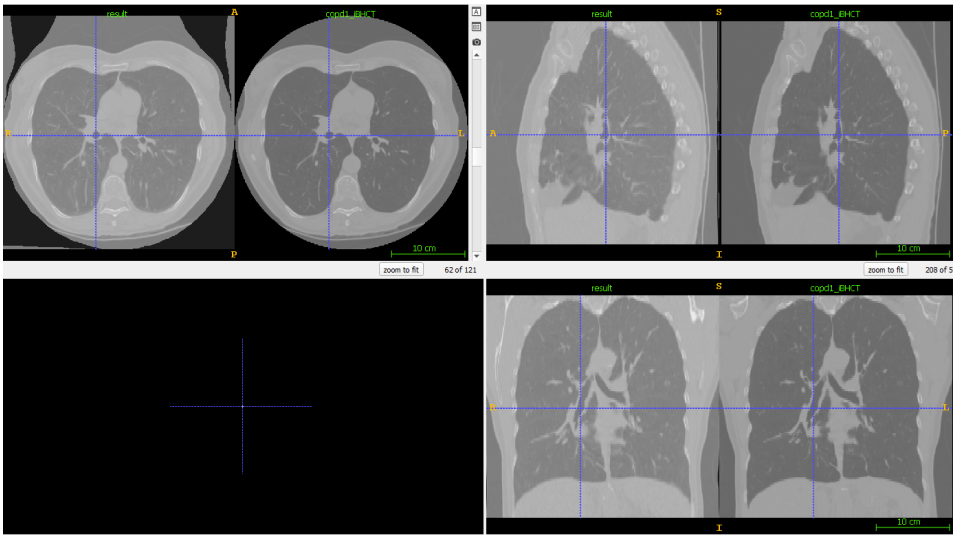


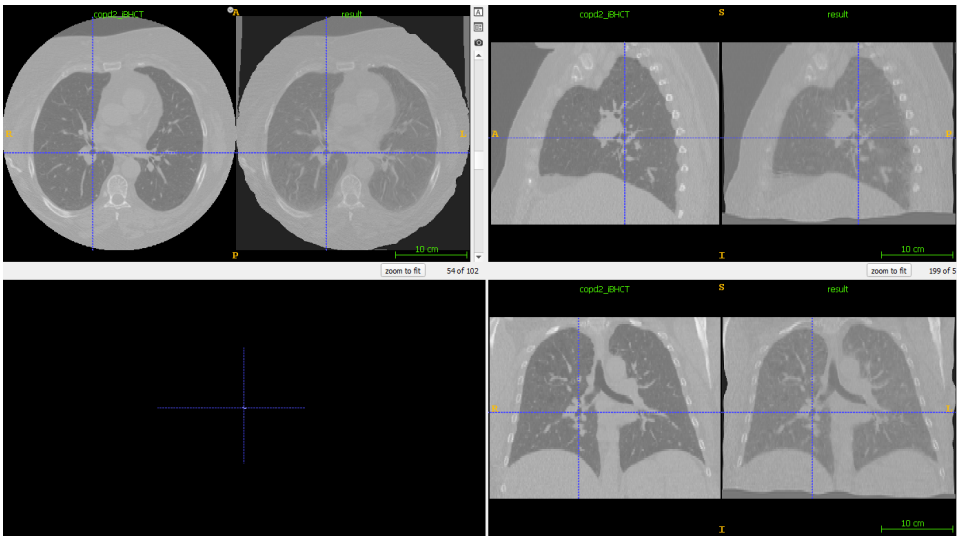
Figure 6: COPDgene4 progress

4.1 Registered Images & Landmarks

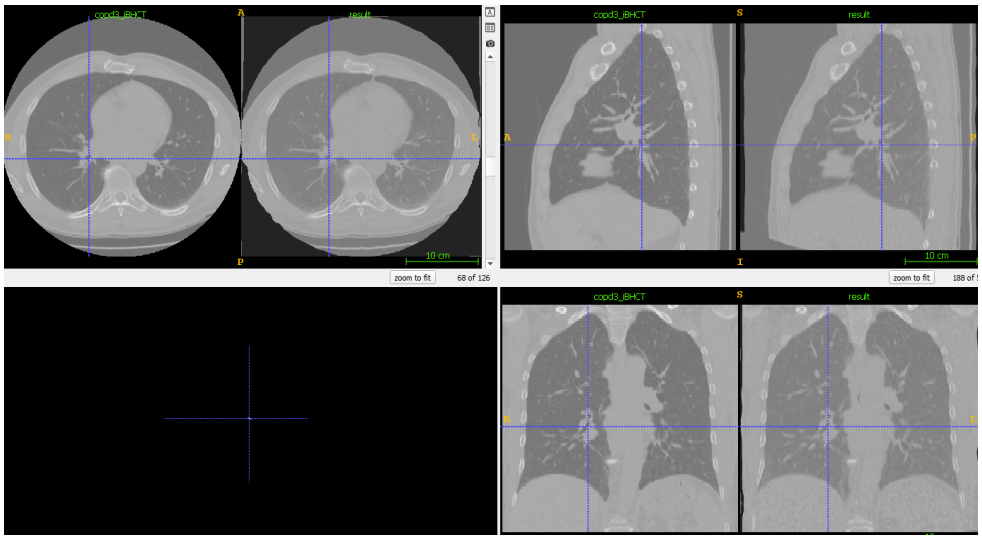
It makes sense to show some representative cases from the registered images. Some slices were selected to be shown here to have an idea about the images under the microscope. Figures 7 and 8 show 4 representative slices from COPDgene (1, 2, 3), and (4) respectively. It can be seen that the cursor on both the original right (fixed inhale) image and on the left images (registered exhale) lies on similar position (kind of a hole).



(a) COPD1, in every corner, the right one is the original inhale image, the left one is the registered exhale



(b) COPD2, in every corner, the right one is the original inhale image, the left one is the registered exhale



(c) COPD3, in every corner, the right one is the original inhale image, the left one is the registered exhale

Figure 7: Showing representative slices of fixed on the right always and registered on the left of axial (top left), sagittal (top right), and coronal (bottom left) of each figure.

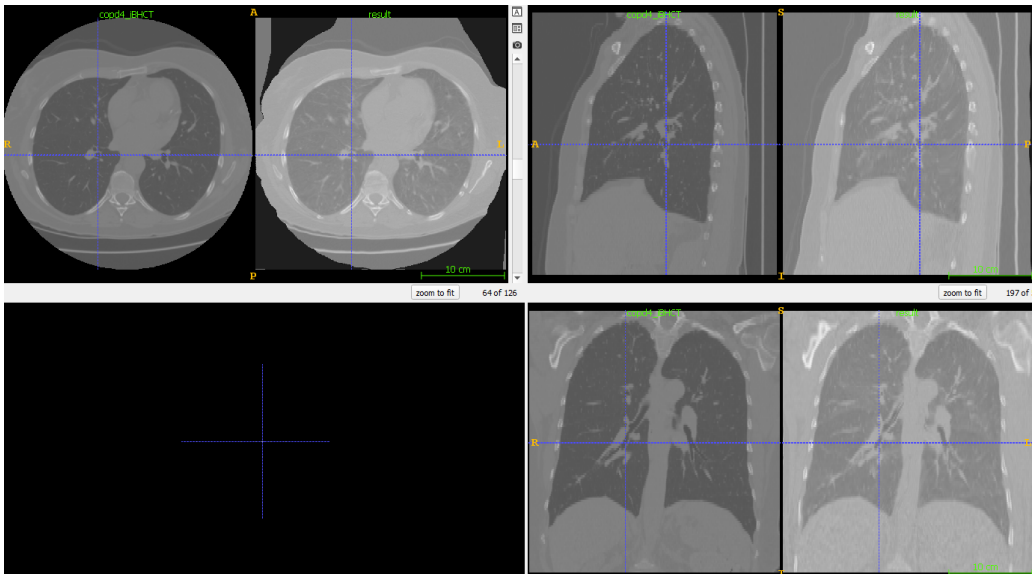


Figure 8: COPD4, in every corner, the right one is the original inhale image, the left one is the registered exhale

Additionally, to show the landmarks overlayed over the exhale image (moving) before registration and after registration, see Figures 9 and 10. Those figures were produced by using the packges provided on dir-lab website (see page 1).

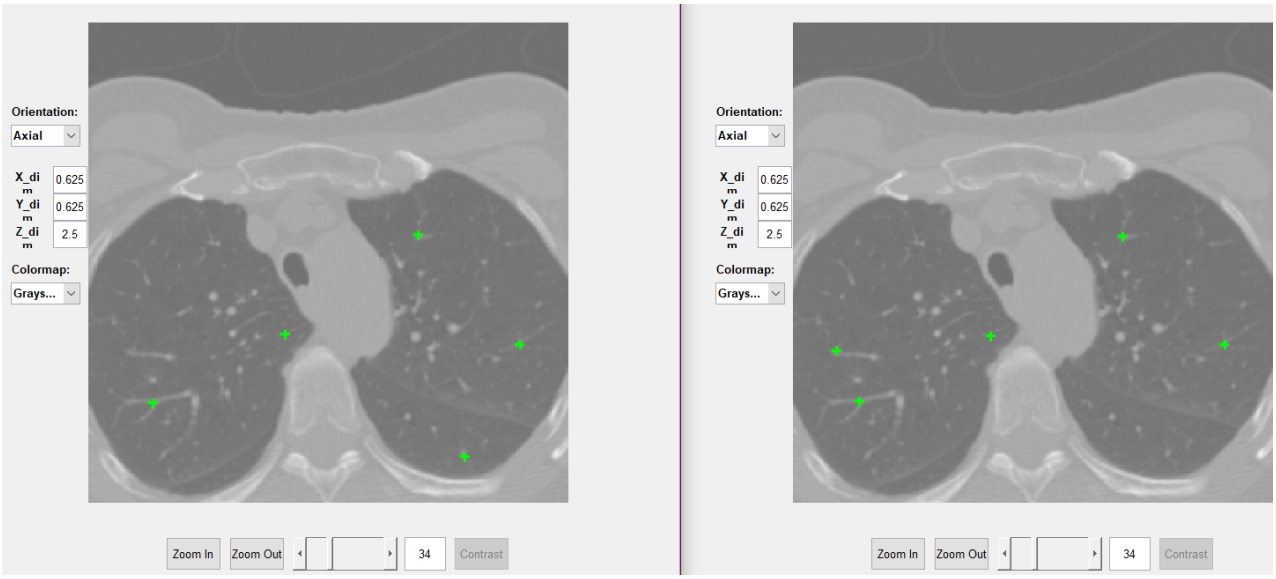
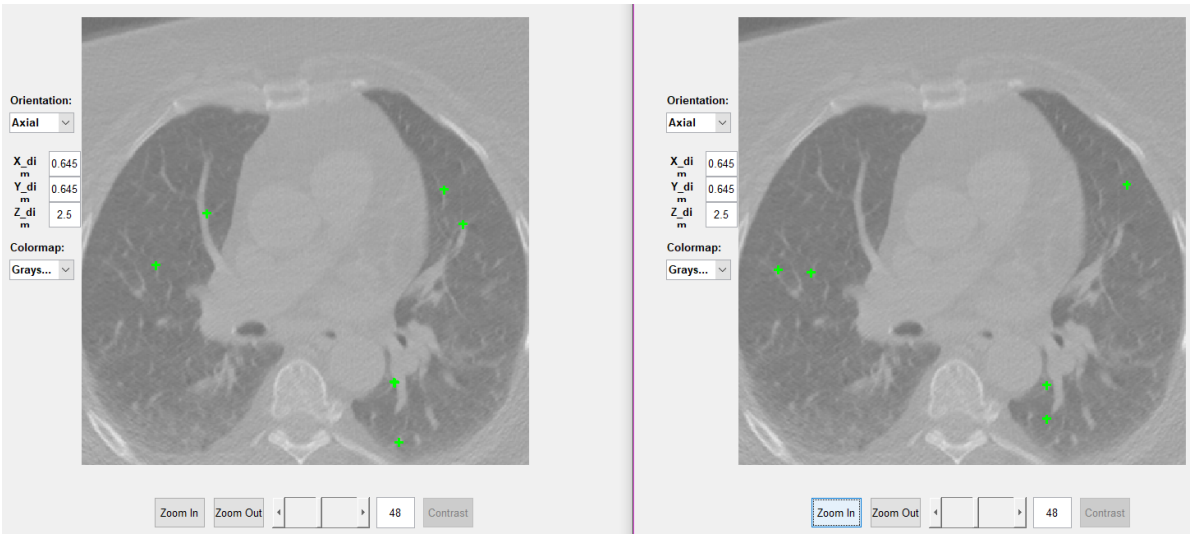
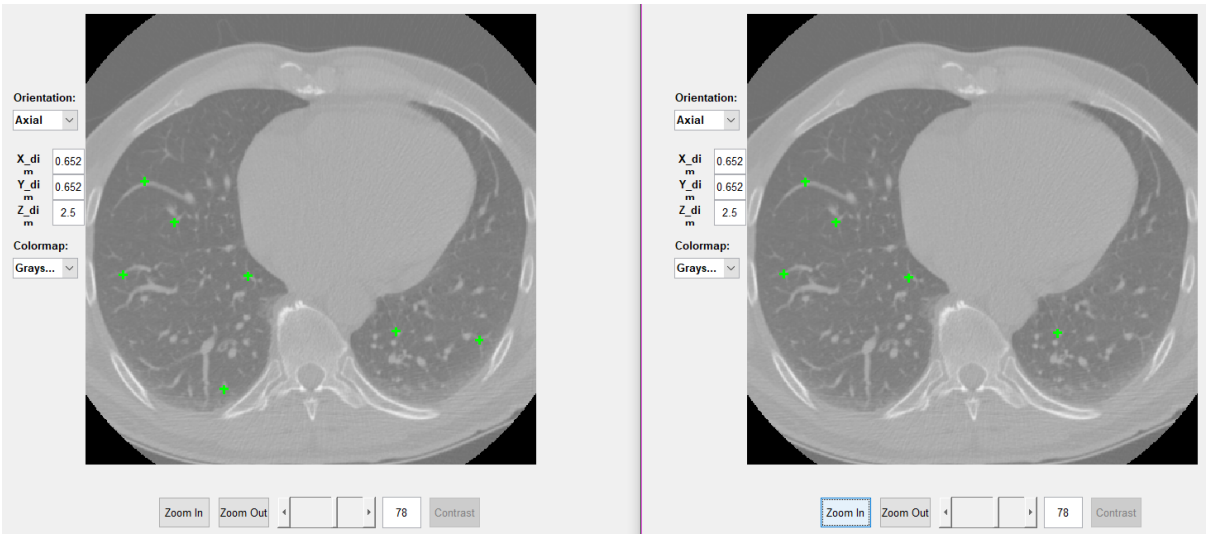


Figure 9: COPD1 exhale image. On the left, the original landmarks, while, on the right, the registered landmarks. The slice shown is 34 axial. Each green crosshair is one landmark.

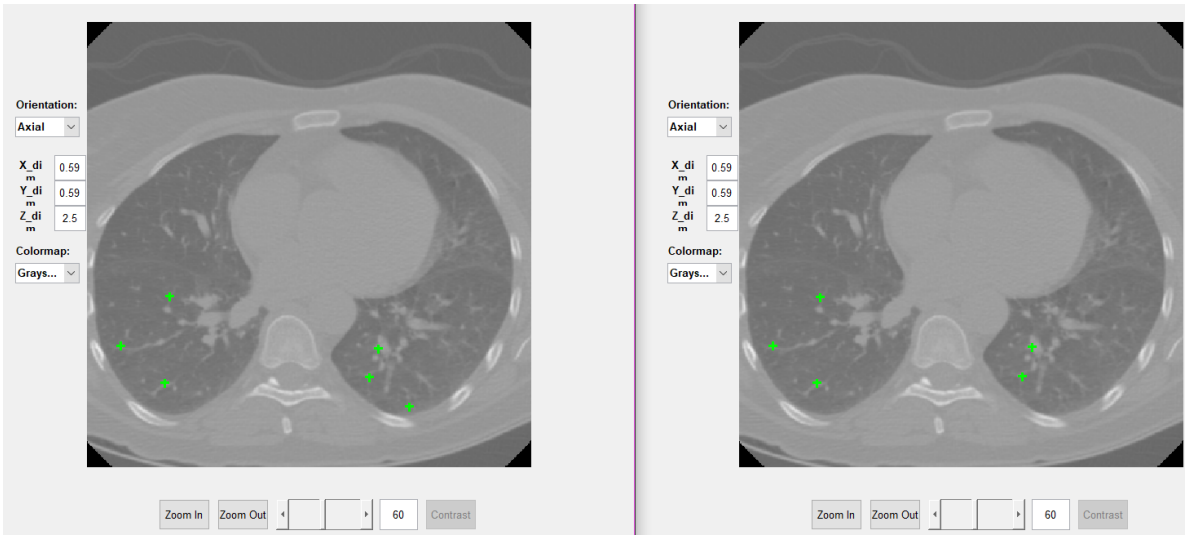
It can be easily seen from Figure 9, that registered landmarks locations are pretty similar to the original ones. Certainly, it is not completely precise. It still have error 1.45 mm with standard deviation 1.7mm which might cause the landmarks to move between axial slices (when the error equals or exceeds 2.5 mm on Z axis). Figure 10 (a) shows that COPD2 has definitely a low registration performance which agrees with numerical values shown in section 4. Figures 10 (b) and (c) show acceptable performance for COPD3 and COPD4 respectively.



(a) COPD2 exhale image. The slice shown is 48 axial.



(b) COPD3 exhale image. The slice shown is 78 axial.



(c) COPD4 exhale image. The slice shown is 60 axial.

Figure 10: showing registered(right) and original (left) landmarks on the axial axis for COPD 2, 3, and 4.

4.2 Time Analysis

All experiment were carried out on a medium-performance machine that has the following specifications:

- Processor: Intel(R) Core(TM) i5 CPU M 520 @ 2.40GHz, 2400 Mhz, 2 Core(s), 4 Logical Processor(s).
- Memory: 6 GB.
- GPU: None.

It is important to have an idea about the duration the registration process of each individual patient needed. 9m 46s, 8m 29s, 8m 21s, 8m 18s were needed to register COPD1, COPD2, COPD3, and COPD4, respectively. Time needed to run `Transformix` was negligible. Additional one minute was required to read and save images.

5 Discussion

It is useful to compare the results got from this project with some published results. NLR [7] is one method puvlished on dir-lab website. Figure 11 shows the comparison between the two methods over the four images. It can be seen that figures are comparable (except for COPD2). So by using only Elastix, we could achieve comparable results to the ones published in internationally-recognized journals.

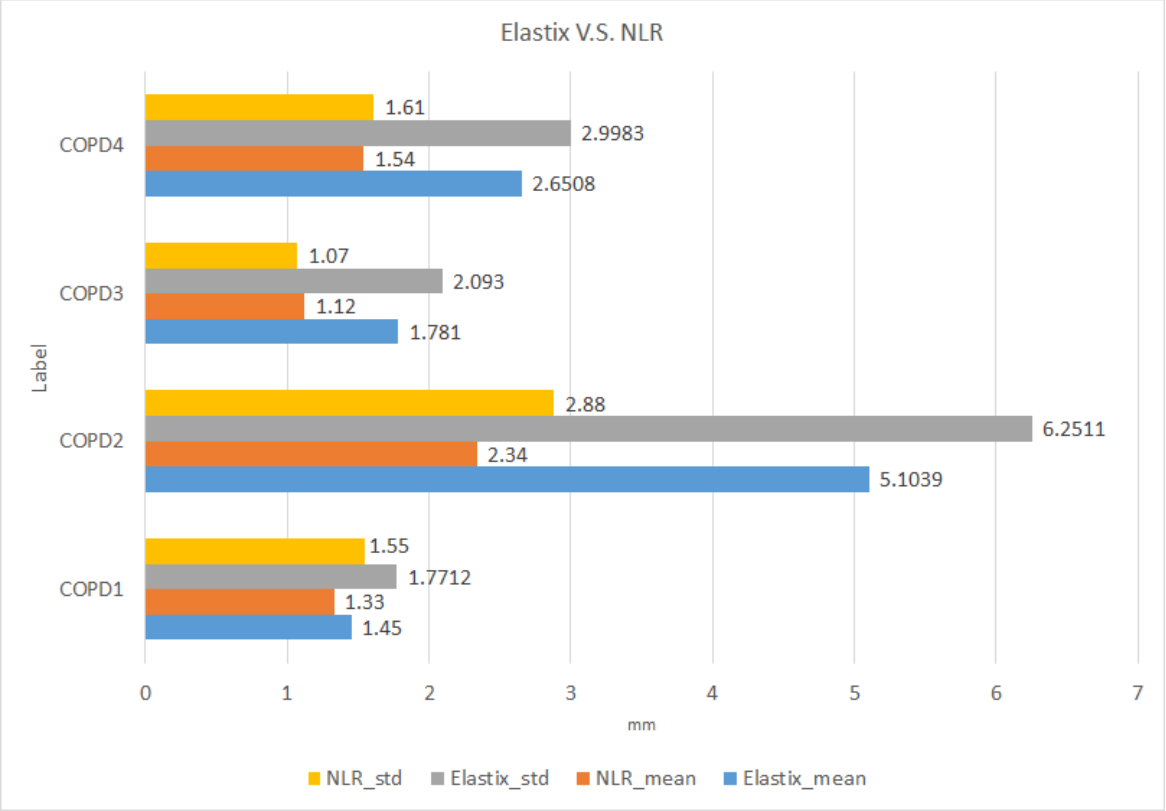


Figure 11: presented Elastix method v.s. NLR [7]

A Final Rigid Parameter File

```
//*********Components
(Registration "MultiResolutionRegistration")
(Metric "AdvancedNormalizedCorrelation")
(ImageSampler "RandomCoordinate")
(Interpolator "LinearInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(Transform "AffineTransform")
(Optimizer "AdaptiveStochasticGradientDescent")
// ********* Pyramids
(NumberOfResolutions 6)//
(FixedImagePyramid "FixedSmoothingImagePyramid")
(MovingImagePyramid "MovingSmoothingImagePyramid")
(ImagePyramidSchedule 14 14 3  10 10 2  8 8 2  4 4 1  2 2 1
1 1 1)//
// ********* Transform
(AutomaticScalesEstimation "true")
(HowToCombineTransforms "Compose")
(AutomaticTransformInitialization "true")//
// ********* Optimizer
(AutomaticParameterEstimation "true")
(ASGDParameterEstimationMethod "DisplacementDistribution")
(UseAdaptiveStepSizes "true")//
(SP_A 50.0)//
// ********* Interpolator
(FinalBSplineInterpolationOrder 3)
//     *********     Data     types     and     formats
(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
(ResultImageFormat "nii.gz")
(ResultImagePixelType "float")
```

(WriteResultImage "false")
// ********* **Image Sampler** (MaximumNumberOfSamplingAttempts 20)
(NumberOfSpatialSamples 7500)//
(UseRandomSampleRegion "true")//
(SampleRegionSize 150.0 150.0 35.0)//
(NewSamplesEveryIteration "true")

Table 4: final rigid registration parameters, a mark // is given at the end of added/changed lines from the initial tables.

B Final Non-Rigid Parameter File

```
//*********Components
(Registration "MultiResolutionRegistration")
(Metric "AdvancedMattesMutualInformation")//
(ImageSampler "RandomCoordinate")
(Interpolator "LinearInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(Transform "BSplineTransform")
(Optimizer "AdaptiveStochasticGradientDescent")

// ********* Pyramids
(NumberOfResolutions 6)//
(FixedImagePyramid "FixedSmoothingImagePyramid")
(MovingImagePyramid "MovingSmoothingImagePyramid")
(ImagePyramidSchedule 14 14 3  10 10 2  8 8 2  4 4 1  2 2 1
1 1 1)//

// ********* Transform
(AutomaticScalesEstimation "true")
(HowToCombineTransforms "Compose")
(AutomaticTransformInitialization "true")
(FinalGridSpacingInVoxels 8.0 8.0 4.0)//

// ********* Optimizer
(AutomaticParameterEstimation "true")//
(UseAdaptiveStepSizes "true")//
(ASGDParameterEstimationMethod "DisplacementDistribution")
(SP_A 50.0)//

// ********* Interpolator
(FinalBSplineInterpolationOrder 3)

// ********* Data types and formats
(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
```

(ResultImageFormat "nii.gz")
(ResultImagePixelType "float")
// ********* **Image Sampler** (MaximumNumberOfSamplingAttempts 20)
(NumberOfSpatialSamples 25000)//
(NewSamplesEveryIteration "true")
(UseRandomSampleRegion "true")//
(SampleRegionSize 150.0 150.0 35.0)//
// ********* **Metric** (NumberOfHistogramBins 64)//

Table 5: final non-rigid registration parameters, a mark // is given at the end of added/changed lines from the initial tables.

References

[1] Isaac Bankman. *Handbook of Medical Imaging: Processing and Analysis*. Academic Press, 2000.

[2] ALEXANDER D. LARS K., JAN R. and JAN L. A matrix-free approach to parallel and memory-efficient deformable image registration. 2018.

[3] K. Murphy M. Viergever S. Klein, M. Staring and J. Pluim. elastix: a toolbox for intensity based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205, 2010.

[4] B.P.F. Lelieveldt M. Smits S. Klein D.P. Shamonin, E.E. Bron and M. Staring. Fast parallel image registration on cpu and gpu for diagnostic classification of alzheimer's disease. *Frontiers in Neuroinformatics*, 7(50):1–15, 2014.

[5] Stefan K. and Marius S. *elastix the manual*.

[6] D. Kim E.A.K. Cohen and R.J. Ober. Cram´er-rao lower bound for point based image registration with heteroscedastic error model for application in single molecule microscopy. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 2015.

[7] Zhang Y Guerrero T. Castillo E, Castillo R. Compressible image registration for thoracic computed tomography images. 29:222–233, 2009.