

# A Survey of Registration Algorithms Based on ITK

Zhen-Huan Zhou\*, Xiao-Jun Wen

School of Computer Engineering, Shenzhen Polytechnic, Shenzhen, China, 518055  
zhouzhenhuan@szpt.edu.cn, wxjun@szpt.edu.cn



Received 3 Jan 2019; Revised 13 Mar 2019; Accepted 13 March 2019

**Abstract.** Medical image registration is becoming more and more important in diagnosis, surgical planning, guiding treatment and monitoring disease progressing, it can align different images to get more valuable information. The programming and development of image registration is rather troublesome, Insight Toolkit (ITK) is a very professional tool, but it is not easy to use. In this paper, first we introduce the ITK architecture and registration framework in detail, and installation under Windows operating system and configuration using python language. Secondly, we introduce the main image registration algorithms of ITK through experiments, including Mattes Mutual Information registration, centered transforms, Multi-Resolution registration, Deformable registration based on FEM and so on. Finally, we give the calculation process and experimental results. With the help of ITK framework, different algorithms are compared and verified, and suitable algorithms are selected for application in engineering, which can shorten the software development cycle and greatly reduce the development cost.

**Keywords:** image registration, ITK, medical image

## 1 Introduction

Insight Segmentation and Registration Toolkit, shortened as the Insight Toolkit (ITK), is an open-source software toolkit for performing registration and segmentation. Although it is large and complex, ITK is designed to be easy to use once you learn about its basic object-oriented and implementation methodology [1-3].

**Table 1.** The big picture

ITK (Medical Image Processing)	Registration / Segmentation	Common	System Basic Pipeline Data
		Basic Filters	PixelWise Neighborhood Global IO
		Algorithms	Registration Markov RF Level Sets Connectedness
		Numerics	Optimizers Statistics VNL FEM

\* Corresponding Author

ITK can be downloaded from the following website: <https://www.itk.org/ITK/resources/software.html>. On the web page, choose the tarball that better fits your system. The options are .zip and .tar.gz files. The first type is better suited for Microsoft-Windows, while the second one is the preferred format for UNIX systems. The data is available from the ITK Web site at the following: <https://www.itk.org/ITK/resources/links.html>. Configuring and Building ITK has been solved through the use of CMake, a cross-platform, open-source build system. CMake can be downloaded from <https://cmake.org/download/>. CMake generates native Makefiles or workspaces to be used with the corresponding development environment of your choice. For example, on UNIX and Cygwin systems, CMake generates Makefiles; under Microsoft Windows CMake generates Visual Studio workspaces. You can download binary versions for Microsoft Windows. Alternatively you can download the source code and build CMake on your system. The minimum version of CMake has been evolving along with the version of ITK. For example, we use the versions of the software as follow:

- (1) Windows 10 Education (x64);
- (2) cmake-3.9.6-win64-x64;
- (3) InsightToolkit-4.13.0;
- (4) Visual Studio 2017;
- (5) Python2.7.3.

## 2 Installing ITK on Microsoft Windows System

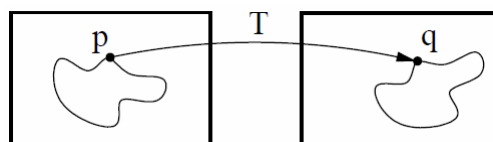
Running CMake to configure and prepare for compilation a new project initially requires two pieces of information: where the source code directory is located, and where the compiled code is to be produced. These are referred to as the *source directory* and the *binary directory* respectively. We recommend setting the binary directory to be different than the source directory in order to produce an *out-of-source* build.

Installation steps as follows:

- (1) Creating ITK source directory on D disk, extracting InsightToolkit-4.13.0 to D:\ITK;
- (2) Creating the binary directory on D disk: D:\ITK\ITK413bin;
- (3) Installing Visual Studio 2015 (2017).
- (4) Running cmake-3.9.6-win64-x64, source file: D:\ITK\InsightToolkit-4.13.0, target file: D:\ITK\ITK413bin, clicking Configure, adding python language option and BUILD\_EXAMPLE, and then Generating (about 30-40 minutes).
- (5) Running itk.sln, right clicking ALL\_BUILD to generate, and right clicking INSTALL to generate. (about 6-10 hours)

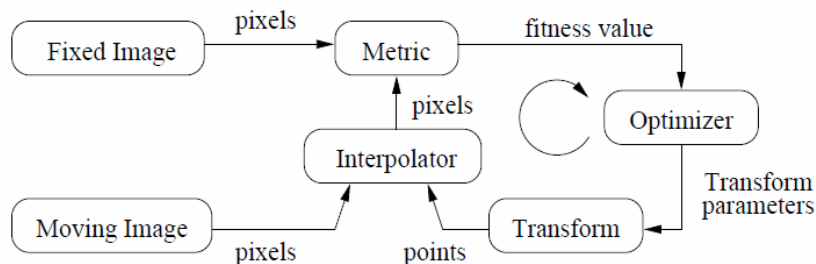
## 3 ITK Registration Framework

Image registration is the process of determining the spatial transform that maps points from one image to homologous points on an object in the second image (Fig. 1).



**Fig. 1.** Image registration is the task of finding a spatial transform mapping one image into another

Let's begin with a simplified typical registration framework where its components and their interconnections are shown in Fig. 2. The basic input data to the registration process are two images: one is defined as the fixed image  $f(x)$  and the other as the moving image  $m(x)$ , where  $x$  represents a position in  $N$ -dimensional space. Registration is treated as an optimization problem with the goal of finding the spatial mapping that will bring the moving image into alignment with the fixed image.

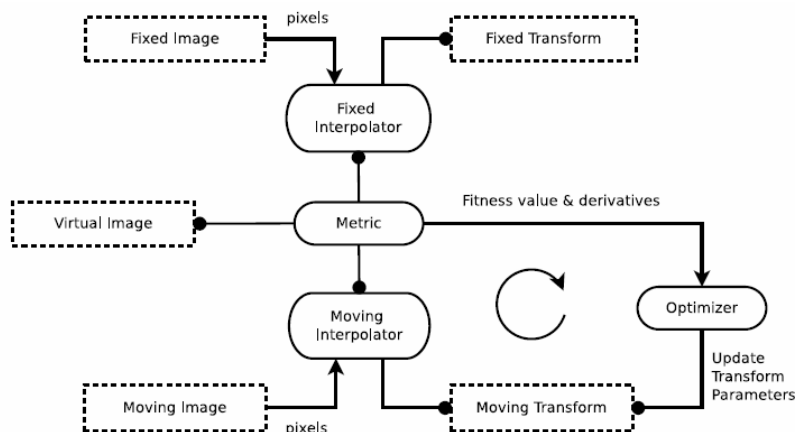


**Fig. 2.** The basic components of a typical registration framework are two input images, a transform, a metric, an interpolator and an optimizer

The transform component  $T(X)$  represents the spatial mapping of points from the fixed image space to points in the moving image space. The interpolator is used to evaluate moving image intensities at non-grid positions. The metric component  $S(f, m \cdot T)$  provides a measure of how well the fixed image is matched by the transformed moving image. This measure forms a quantitative criterion to be optimized by the optimizer over the search space defined by the parameters of the transform [4-6].

Registration framework provides more flexibility to the above traditional registration concept. In this new framework, the registration computations can happen on a physical grid completely different than the fixed image domain having different sampling density.

Various registration components are illustrated in Fig. 3. Boxes with dashed borders show data objects, while those with solid borders show process objects. The matching Metric class is a key component that controls most parts of the registration process since it handles fixed, moving and virtual images as well as fixed and moving transforms and interpolators [7-8].



**Fig. 3.** The basic components of the registration framework

Fixed and moving transforms and interpolators are used by the metric to evaluate the intensity values of the fixed and moving images at each physical point of the virtual space. Those intensity values are then used by the metric cost function to evaluate the fitness value and derivatives, which are passed to the optimizer that asks the moving transform to update its parameters based on the outputs of the cost function. Since the moving transform is shared between metric and optimizer, the above process will be repeated till the convergence criteria are met [9-10].

#### 4 Registering Two Images Using Registration Framework

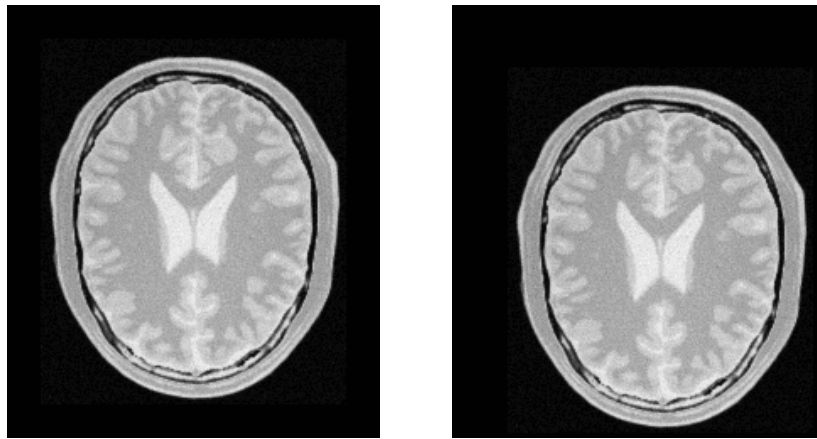
A registration method requires the following set of components: two input images, a transform, a metric and an optimizer. This example illustrates the use of the image registration framework in Insight. It should be read as a “Hello World” for ITK registration. Instead of means to an end, this example should be read as a basic introduction to the elements typically involved when solving a problem of image registration. In order to register the two images: BrainProtonDensitySliceBorder20.png, BrainProton

DensitySliceShifted13x17y.png. You can copy the two input images in the directory of D:\ITK\ITK413bin\bin\Debug, execute the following command at the cmd terminal:

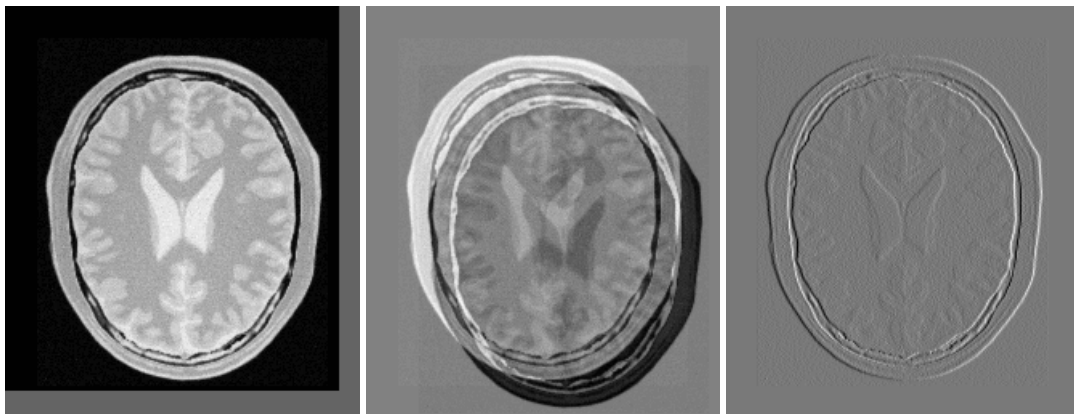
```
D:\ITK\ITK413bin\bin\Debug>ImageRegistration1.exe BrainProtonDensitySliceBorder20.png BrainProtonDensitySliceShifted13x17y.png ImageRegistration1Output.png ImageRegistration1DifferenceAfter.png
```

Here, BrainProtonDensitySliceBorder20.png is fixed image and BrainProtonDensitySliceShifted13x17y.png is moving image, they are input images. ImageRegistration1Output.png, ImageRegistration1DifferenceAfter.png, ImageRegistration1DifferenceBefore.png are three output images.

The second image is the result of intentionally translating the first image by (13, 17) millimeters. Both images have unit-spacing and are shown in Fig. 4. The registration takes 22 iterations and the resulting transform parameters are in the Fig. 6.



**Fig. 4.** BrainProtonDensitySliceBorder20.png, BrainProtonDensitySliceShifted13x17y.png



**Fig. 5.** ImageRegistration1Output.png, ImageRegistration1DifferenceAfter.png, ImageRegistration1DifferenceBefore.png

```

0 = 4499.45 : [2.9286959512455857, 2.7244705953923805]
1 = 3860.84 : [6.135143776902401, 5.115849348610004]
2 = 3508.02 : [8.822660051952475, 8.078492808653918]
3 = 3117.31 : [10.968558473732326, 11.454158663474674]
4 = 2125.43 : [13.105290365964755, 14.835634202454191]
5 = 911.308 : [12.75173580401588, 18.819978461140323]
6 = 741.417 : [13.139053510563274, 16.857840597942413]
7 = 16.8918 : [12.356787624301035, 17.480785285045815]
8 = 233.714 : [12.79212443526829, 17.234854683011704]
9 = 39.8027 : [13.167510875734614, 16.904574468172815]
10 = 16.5731 : [12.938831371165355, 17.005597654570586]
11 = 1.68763 : [13.063495692092735, 16.996443033457986]
12 = 1.79437 : [13.001061362657559, 16.999307384689935]
13 = 0.000762481 : [12.945418587211314, 17.0277701944711]
14 = 1.74802 : [12.974454390534774, 17.01621663980765]
15 = 0.430253 : [13.002439510423766, 17.002309966416835]
16 = 0.00531816 : [12.989877586882951, 16.99301810428082]
17 = 0.0721346 : [12.996759235073881, 16.996716492365685]
18 = 0.00996773 : [13.00288423694971, 17.00156618393022]
19 = 0.00516378 : [12.99928608126834, 17.000045636412015]
20 = 0.000228075 : [13.00123653240422, 16.999943471681494]
Optimizer stop condition: RegularStepGradientDescentOptimizeru4: Step too small
after 21 iterations. Current step (0.000976563) is less than minimum step (0.001
).
Registration done !
Number of iterations = 22
Translation along X = 13.0012
Translation along Y = 16.9999
Optimal metric value = 0.000679396
D:\ZZH\UTK\ITK\itkbin411\bin\Debug>

```

Fig. 6. The results of iteration and registration

### 5 Mattes Mutual Information Registration

Some of the most challenging cases of image registration arise when images of different modalities are involved. In such cases, metrics based on direct comparison of gray levels are not applicable. It has been extensively shown that metrics based on the evaluation of mutual information are well suited for overcoming the difficulties of multi-modality registration. The concept of Mutual Information is derived from Information Theory and its application to image registration has been proposed in different forms. The Insight Toolkit currently provides the implementations of Mutual Information metrics.

BrainT1SliceBorder20.png is a T1 MRI (fixed image) and BrainProtonDensitySliceShifted13x17y.png is a proton density MRI (moving image) (Fig. 7). ImageRegistration4Output.png, ImageRegistration4CheckerboardBefore.png, ImageRegistration4CheckerboardAfter.png are output images.

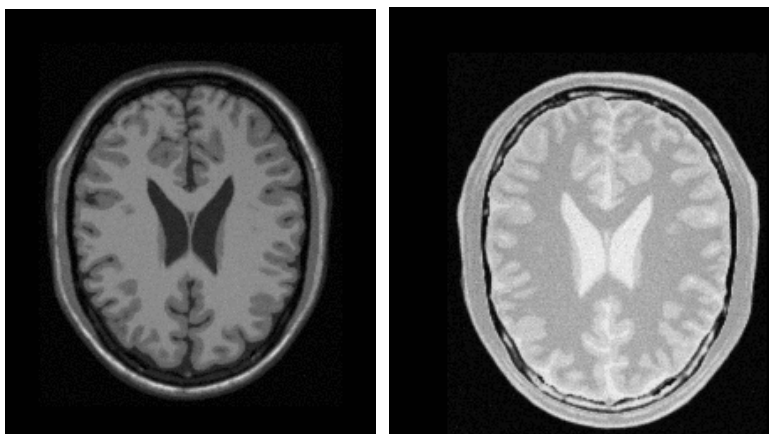


Fig. 7. BrainT1SliceBorder20, BrainProtonDensitySliceShifted13x17y.png

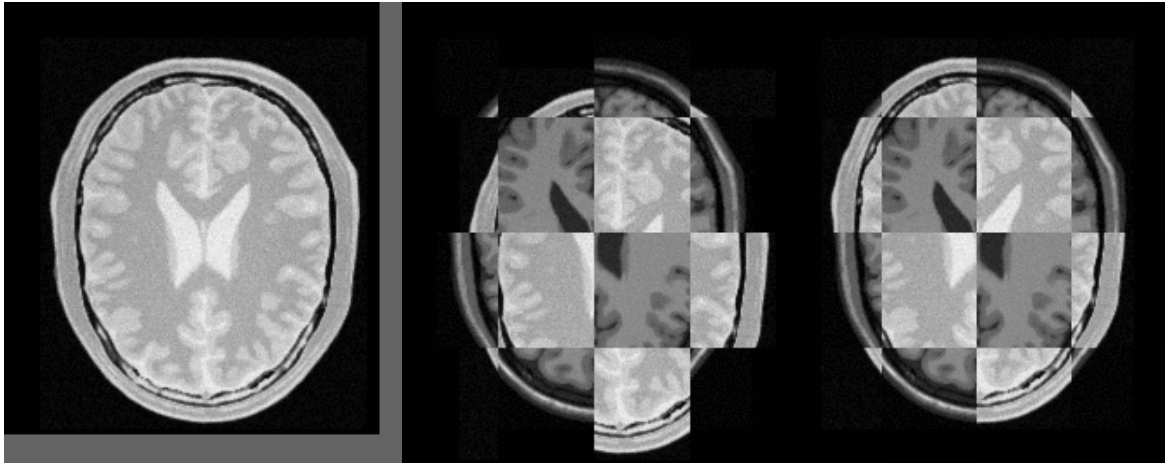


Fig. 8. ImageRegistration4Output.png, ImageRegistration4CheckerboardBefore.png, ImageRegistration4CheckerboardAfter.png are output images

```

管理员: C:\Windows\system32\cmd.exe
D:\ZZH\UTK\ITK\itkbin411\bin\Debug>ImageRegistration4.exe BrainT1$liceBorder20.p
ng BrainProtonDensitySliceShifted13x17y.png ImageRegistration4Output.png 100 Ima
geRegistration4CheckerboardBefore.png ImageRegistration4CheckerboardAfter.png 24

0  -0.326452  [5.601180396185191, 5.711985483996847]
1  -0.395374  [10.732693433691866, 11.849376945024815]
2  -0.561712  [16.522977275400088, 17.369577397223328]
3  -0.617083  [10.122995086301847, 17.354478416287893]
4  -0.663853  [15.226196137601852, 16.94006433591593]
5  -0.718399  [11.13019953463754, 16.945339608879095]
6  -0.756991  [14.404679566206937, 17.068622871543937]
7  -0.810539  [11.783262276488156, 17.07953464332674]
8  -0.83739   [13.879834722218167, 17.12883465255271]
9  -0.875711  [12.203937592644458, 17.050613311518646]
10 -0.896797  [13.546094678784002, 17.057975856839946]
11 -0.923412  [12.477930844150048, 16.94867111317831]
12 -0.932824  [13.328475003218275, 17.068855955913605]
13 -0.952735  [12.645591436638181, 16.992001009541774]
14 -0.954076  [13.19299872820298, 17.04276233384827]
15 -0.967352  [12.754552829893992, 17.008217791669136]
16 -0.964563  [13.10484788481901, 17.041193213176932]
17 -0.973847  [12.825068066870987, 17.010348181433788]
18 -0.969744  [13.048365101361421, 17.039407747791646]
19 -0.9762    [12.869740100977783, 17.016063270847667]
20 -0.972572  [13.012692516851095, 17.034333307249465]

```

Fig. 9. Results of Mattes Mutual Information Registration (1)

```

管理员: C:\Windows\system32\cmd.exe
21 -0.976997 [12.897667203870542, 17.02649383459021]
22 -0.974183 [12.98959727179727, 17.03397192151205]
23 -0.977056 [13.046180690703345, 17.106809932728197]
24 -0.97575 [12.990360442797176, 17.05855417872488]
25 -0.977222 [12.983334166726525, 16.985102498585388]
26 -0.976261 [12.998138952401066, 17.04224538754282]
27 -0.977127 [12.958610819498771, 17.01640740718228]
28 -0.976366 [12.993546779286826, 17.030785390573335]
29 -0.977038 [12.967984969957294, 17.046910911561273]
30 -0.976884 [12.991126274366177, 17.039905150704108]
31 -0.977108 [12.983876374397004, 17.021972404969176]
32 -0.9769 [12.993247567192933, 17.03428634722346]
33 -0.977068 [12.980868760359439, 17.034165122367906]
34 -0.976987 [12.990726914367203, 17.03511196669229]
35 -0.97707 [12.98353080493895, 17.03179729831514]
36 -0.976994 [12.989522910792443, 17.0338632408475]
37 -0.977055 [12.99352885226736, 17.038775051512214]
38 -0.977104 [12.989538549371822, 17.035646395009813]
39 -0.97707 [12.991847387757057, 17.0311319433245]
40 -0.977039 [12.989198738023173, 17.03420435282852]
41 -0.977056 [12.9921845837846, 17.035475554722726]
42 -0.977076 [12.989707830657037, 17.0346972998725]
Optimizer stop condition: RegularStepGradientDescentOptimizerV4: Gradient magnit
ude tolerance met after 43 iterations. Gradient magnitude (9.46776e-005) is less
than gradient magnitude tolerance (0.0001).

Result =
Translation X = 12.9833
Translation Y = 16.9851
Iterations = 44
Metric value = -0.977222
Stop Condition = RegularStepGradientDescentOptimizerV4: Gradient magnitude tol
erance met after 43 iterations. Gradient magnitude (9.46776e-005) is less than g
radient magnitude tolerance (0.0001).

D:\ZZH\UTK\ITK\itkbin411\bin\Debug>

```

Fig. 10. Results of mattes mutual information registration (2)

## 6 Centered Transforms

The ITK image coordinate origin is typically located in one of the image corners. This results in counter-intuitive transform behavior when rotations and scaling are involved. Users tend to assume that rotations and scaling are performed around a fixed point at the center of the image.

BrainProtonDensitySliceBorder20.png is a fixed image, BrainProtonDensitySliceR10X13Y17S12.png is a moving image, and the second image is the result of intentionally rotating the first image by 10 degrees, scaling by 1/1.2 and then translating by (13, -17). Both images have unit-spacing and are shown in Fig.11.

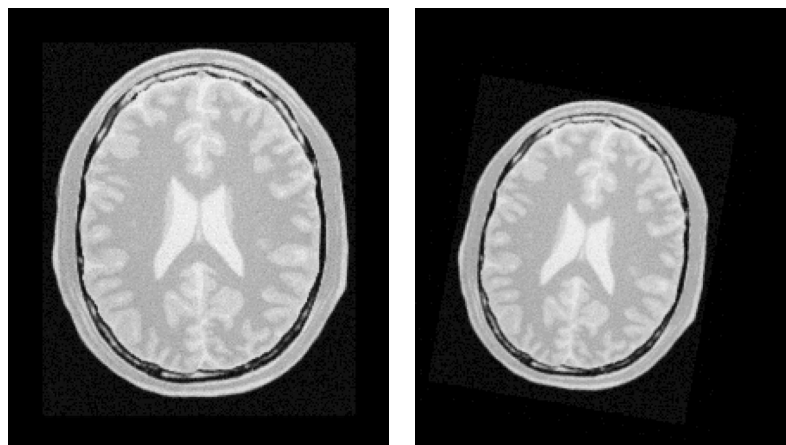
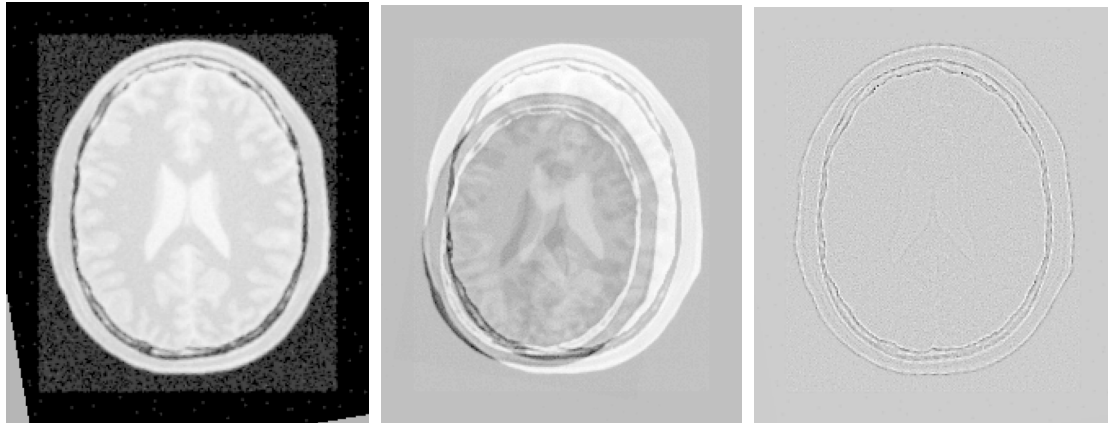


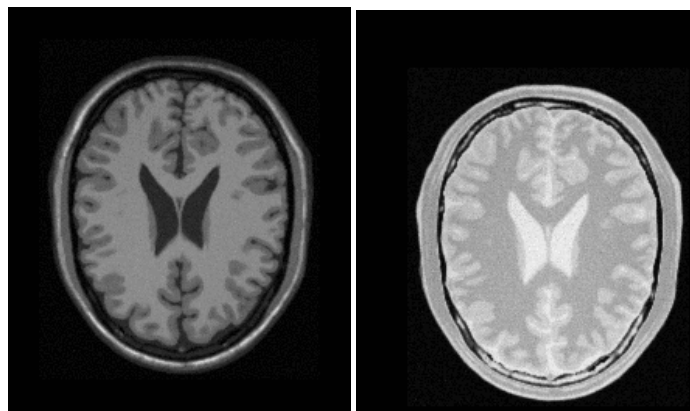
Fig. 11. Fixed and moving images are provided as input to the registration method using the Similarity2Dtransform



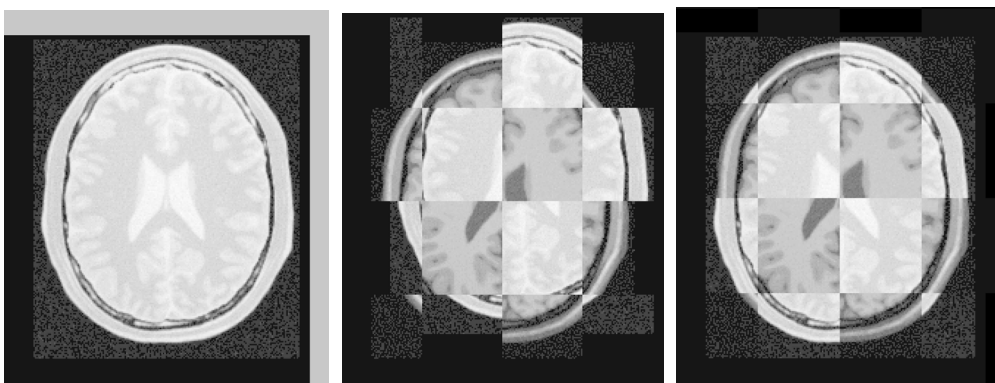
**Fig. 12.** Resampled moving image (left). Differences between fixed and moving images, before (center) and after (right) registration with the Similarity2D transform

## 7 Multi-Resolution Registration

Performing image registration using a multi-resolution approach is widely used to improve speed, accuracy and robustness. The basic idea is that registration is first performed at a coarse scale where the images have fewer pixels. The spatial mapping determined at the coarse level is then used to initialize registration at the next finer scale. This process is repeated until it reaches the finest possible scale. This coarse-to-fine strategy greatly improves the registration success rate and also increases robustness by eliminating local optima at coarser scales. Robustness can be improved even more by smoothing the images at coarse scales.



**Fig. 13.** BrainT1SliceBorder20.png and BrainProtonDensitySliceShifted13x17y.png are the fixed and moving images



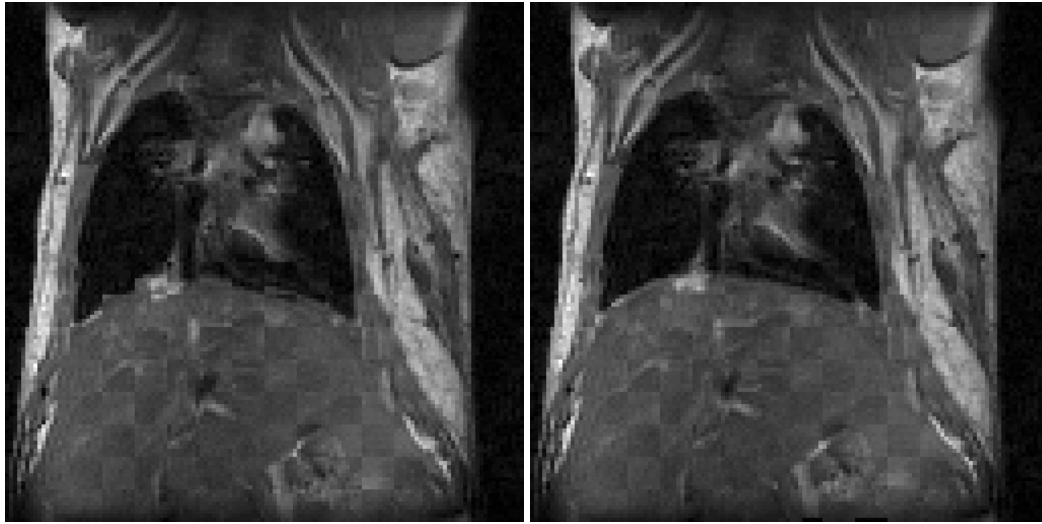
**Fig. 14.** Mapped moving image (left) and composition of fixed and moving images before (center) and after (right) registration



## 8 Deformable Registration Based on FEM

The finite element (FEM) library within the Insight Toolkit can be used to solve deformable image registration problems.

Fig. 15 presents the results of the FEM-based deformable registration applied to two time separated slices of a living rat dataset. Checkerboard comparisons of the two images are shown before registration (left) and after registration (right). Both images were acquired from the same living rat, the first after inspiration of air into the lungs and the second after exhalation. Deformation occurs due to the relaxation of the diaphragm and the intercostal muscles, both of which exert force on the lung tissue and cause air to be expelled.



**Fig. 15.** Checkerboard comparisons before and after FEM-based deformable registration

## 9 Conclusions

Medical image is different from general images, it has a special image format and processing methods, in order to pursue speed and real-time, people generally choose C++ programming language, ITK is an open source C++ library of medical image segmentation and registration. But even for a skilled programmer, programming in C++ to process medical images is not an easy task, so ITK provides the python language source code to help beginners get started faster. In addition, there are more and more image registration algorithms, including not only 2D images, 3D images, but also point set registration. These registration methods can be applied not only in medicine, but also in many other fields. The future work is to make more and more people familiar with and use ITK, and add more algorithms for ITK to make it more perfect. Reduce researchers' programming time and shorten the research and development cycle of medical image research.

## References

- [1] T.S. Yoo1, M.J. Ackerman, W.E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, R. Whitaker, Engineering and algorithm design for an image processing API: a technical report on ITK - the insight toolkit, in: J.D. Westwood, H.M. Hoffman, R.A. Robb, D. Stredney (Eds.), Studies in Health Technology and Informatics, Vol. 85., IOS Press, Amsterdam, 2002, pp. 586-592.
- [2] R. Shams, P. Sadeghi, A survey of medical image registration on multicore and the GPU, IEEE Signal Processing Magazine 27(2)(2010) 50-60.
- [3] J.P. W. Pluim, J.B. Antoine, M.A. Viergever, Mutual information based registration of medical images: a survey, IEEE

- Transactions on medical imaging 22(8)(2003) 986-1004.
- [4] J.B.A. Maintz, M.A. Viergever, A survey of medical image registration, *Medical Image Analysis* 2(1)(1998) 1-36.
  - [5] K. Murphy, B. van Ginneken, Semi-automatic construction of reference standards for evaluation of image registration, *Medical Image Analysis* 15(2011) 71-84.
  - [6] M. Urschler, S. Kluckner, H. Bischof, A framework for comparison and evaluation of nonlinear intra-subject image registration algorithms, in: *Proc. MICCAI Workshop on Open Science*, 2007.
  - [7] T. Vik, S. Kabus, J. von Berg, K. Ens, S. Dries, T. Klinder, C. Lorenz, Validation and comparison of registration methods for free breathing 4D lung CT, in: *Proc. the SPIE* 6917, 2008.
  - [8] R. Wiemker, B. deHoop, S. Kabus, H.A. Gietema, R. Opfer, E. Dharaiya, Performance study of a globally elastic locally rigid matching algorithm for follow-up chest CT, in: *Proc. the SPIE* 6917, 2008.
  - [9] V. Chu, G. Hamarneh, MATLAB-ITK interface for medical image filtering, segmentation, and registration, in: *Proc. the SPIE* 6144, 2006.
  - [10] H. Zhang, F. Yang, Multimodality medical image registration using hybrid optimization algorithm, in: *Proc. 2008 International Conference on BMEI*, 2008.