# MultiResMIRegistration

This command line application demonstrates the use of ITK components for performing rigid multi-modal registration using mutual information.

## *What is Mutual Information?*

Mutual information measure how much information one random variable tells about another. The use of mutual information for medical image registration applications was independently introduced in 1995 by Viola and Wells [1] and Collignon[2]. For two images, the mutual information is computed from the joint probability distribution of the images' intensity or gray-values. When two images are aligned, the joint probability distribution is "peaky" resulting in a high mutual information value. Mis-registration causes the distribution to disperse resulting in a low mutual information value.

One of the main advantages of using mutual information is than it can be used to align images of different modalities (e.g. CT to MR-T1, MR-T1 to PET etc).

More information on mutual information based registration can be found at the following web sites:

- Paul Viola's homepage: contains links to his Ph.D thesis, papers and pointer to other mutual information resources
  `http://www.ai.mit.edu/people/viola/viola.html`

- Sebastien Gilles's web tutorial:
  `http://www-rocq.inria.fr/~gilles/IMMMI/`

- This page from the Image Sciences Institute of Utrecht University has an animation of the effect of mis-registration on the joint probability distribution:
  `http://www.isi.uu.nl/Research/Registration/registration-project.html`

## *Application overview:*

The application reads in two 3D raw image volumes: the target volume and the reference volume. The application then iteratively estimates the rigid transform that will align the reference onto the target volume.

The application terminates after a completing a user-defined number of iterations. The estimate rigid transform is applied to the reference, the resulting registered image is then written out to file as a raw 3D image volume.

Optionally, each 2D slice form the target, reference and registered image volume can be written out as PGM files – facilitating viewing with simple 2D image viewers.

## *What components of ITK does this application use?*

The application makes use of the ITK registration framework and ITK multi-resolution framework.

### ITK Registration framework

The ITK registration framework is a generic framework for registering images and point sets. A registration algorithm is built from three components: a `Transform`, a `Metric` and an `Optimizer`. Standard component API's allowing mixing and matching of the components.

This application uses the `itk::QuaternionRigidRegistrationTransform`, the `itk::MutualInformationImageToImageMetric` and the `itk::GradientDescentOptimizer`.

### ITK Multi-resolution Registration framework

Performing image registration using a multi-resolution strategy has been widely shown to improve speed, accuracy and robustness. The ITK Multi-resolution registration framework is a generic framework for defining a multi-resolution registration scheme. There are two major components in the framework: the underlying registration method and the down-sampling strategy.

Any `RegistrationMethod` that meets the minimal API requirements can plug into the multi-resolution framework. The down-sampling strategy is encapsulated in a `MultiResolutionImagePyramid` object. A flexible schedule-based scheme allows the user to define the shrink factors at each level of the pyramid.

The multi-resolution scheme used in this application is define in `itk::MultiResolutionMutualInformationRigidRegistration`.

## *How do I run the application?*

The application takes one argument: the name of a parameter file. A valid parameter file has the following format:

```
N:/ImageData/target.raw
1
```

```
256 256 26
1.25 1.25 4.0
N:/ImageData/reference.raw
1
512 512 29
0.653595 0.653595 4.0
5
4 4 1
8 8 1
2500 2500 2500 2500 2500
1e-4 1e-5 5e-6 1e-6 5e-7
320 320 320 320 320
registered.raw
1
1
pgmsdir
```

The first entry is the name of the file containing the raw 3D target volume. The second entry specifies the endian-ness of the file: a non-zero number represent big-endian and zero represent little-endian. The next three numbers specifies the size of  volume in column, row, slice order. The following three numbers is the target volume pixel spacing also in column, row, slice order.

The ninth entry is the name of the file containing the raw 3D reference volume. The tenth entry specifies the endian-ness. The next six numbers specifies the size and the pixel spacing of the volume.

Both input volumes are assumed to be in binary (signed short) format.

The seventeenth entry is the number multi-resolution level to be used (5 in this case). The next three numbers specifies the starting (level-0) shrink factors for the target volume. The following three numbers specifies the starting shrink factors for the reference volume. The shrink factors for all other level are computed automatically by dividing the shrink factors of the previous level by a factor of 2. All shrink factors less than 1 is round up to 1.

The next 5 numbers specifies the number of registration iterations to be performed at each multi-resolution level. The next 5 numbers specifies the learning at each level. The next 5 numbers after that specifies the translation scale at each level.

The 39$^{th}$ and 40$^{th}$ entries specify the output filename and endian-ness for the registered image volume.

The 41$^{st}$ entry switches on/off the PGM file dumping option: a non-zero number turns the PGM option on, a zero turns the PGM option off.

If the PGM option is turned on, the 42$^{nd}$ entry specifies the directory where the output PGM files are to be written (NB: the directory must already exist otherwise no images are written out). Slices from the target volume are prefixed `target` then followed by the slice number starting from `000`. Similarly, slices from the reference volume are prefixed `reference` and slices from the registered image are prefixed `register`.

## Where can I get some input data?

The Vanderbilt Retrospective Registration Evaluation Project contains multi-modal images (CT, MR-T1, MR-T2, MR-PD, PET) of a dozen patients.

The following results were produced using parameter file PracCTToT1.txt.

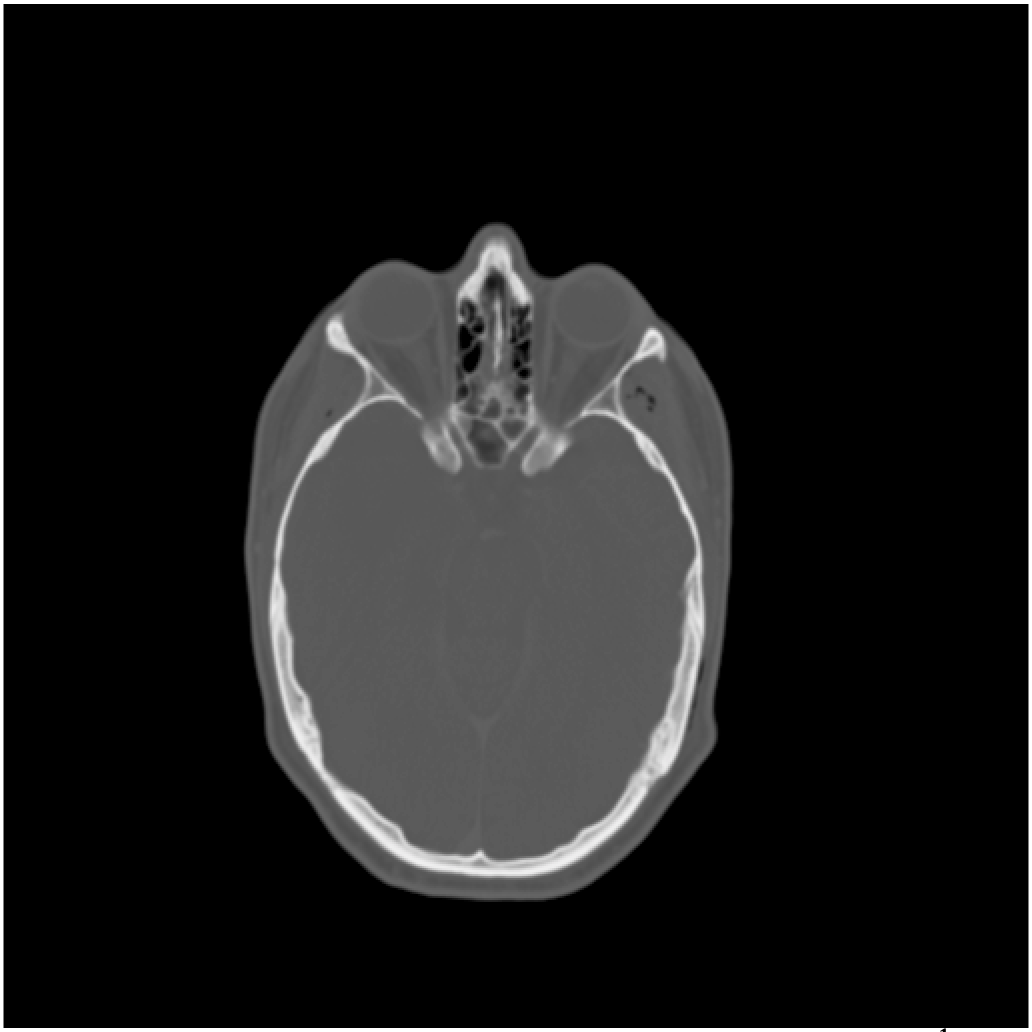

**Fig 1: Slice 7 from the practice patient CT volume (the reference volume)[1]**

[1] The images were provided as part of the project, ``Evaluation of Retrospective Image Registration'', National Institutes of Health, Project Number 1 R01 NS33926-01, Principal Investigator, J. Michael Fitzpatrick, Vanderbilt University, Nashville, TN.

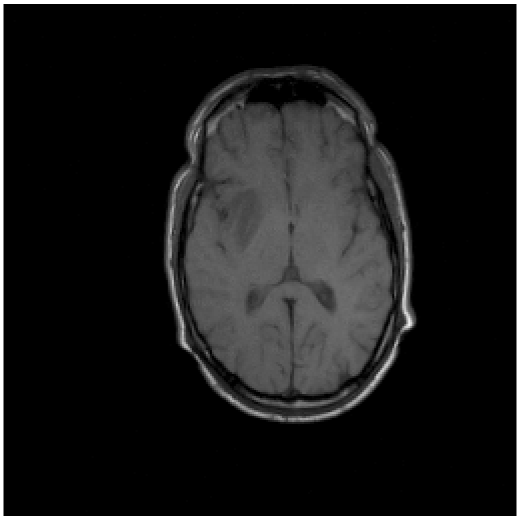

**Fig 2: Slice 7 from the practice patient MR-T1 volume (the target volume)**

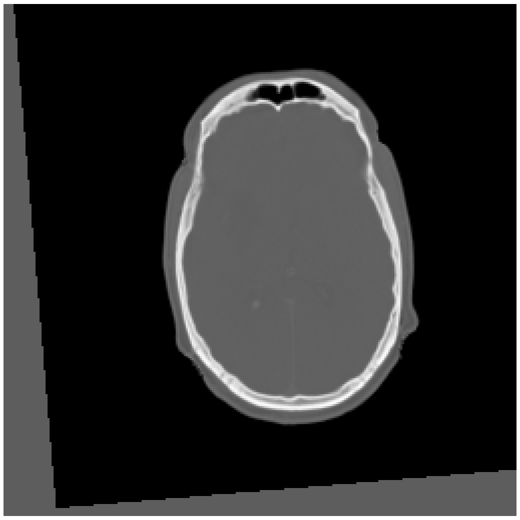

**Fig 3: Slice 7 from the CT-to-MR-TI registered image**

The following results were produced using parameter file PracPETToPD.txt.

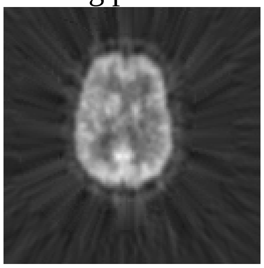

**Fig 4: Slice 7 from the practice patient PET volume (the reference volume)**

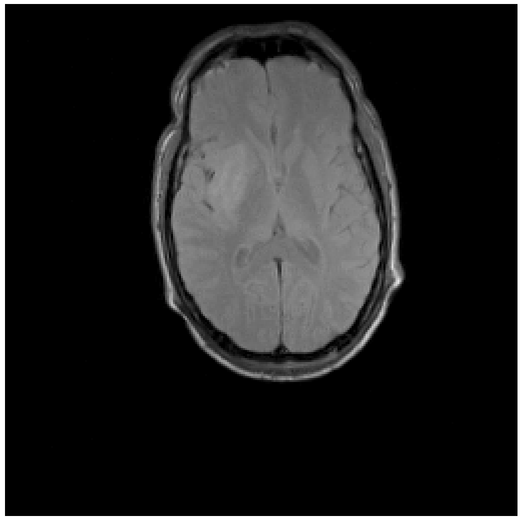

**Fig 5: Slice 7 from the practice patient MR-PD volume (the target volume)**

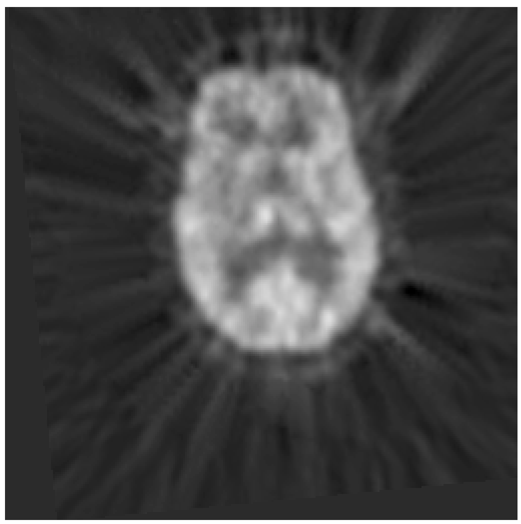

**Fig 6: Slice 7 from the PET-to-MR-PD registered image**

## *References*

[1] P. Viola and W.M. Wells III, "Alignment by maximization of mutual information", *International Conference on Computer Vision* (E. Grimson, S. Shafer, A. Blake and K. Sugihara, eds.), IEEE Computer Society Press, Los Alamitos, CA, pp. 16-23, 1995.

[2] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens and G. Marchal, "Automated multi-modality image registration based on information theory", *Information Processing in Medical Imaging* (Y. Bizais, C. Barillot and R. Di Paola, eds.), Kluwer Academic Publishers, Dordrecht, pp. 263-274, 1995.