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 is a measure of 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 both 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 pointers 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:

This application reads in two 3D raw image volumes: the fixed (target) volume and the moving (source) volume. The application then iteratively estimates the rigid transform that will align the moving onto the fixed volume.

The application terminates after completing a user-defined number of iterations. The estimated rigid transform is applied to the moving/source image. Each 2D slice from the fixed, moving and registered image volume is written out as PGM files – facilitating viewing with simple 2D image viewers.

What components of ITK does this application use?

This 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. A registration algorithm is built from several components: a `Transform`, an `Interpolator`, a `Metric` and an `Optimizer`. Standard component API's allows mixing and matching of the components.

The part of the registration framework used by this application are the `itk::QuaternionRigidRegistrationTransform`, the `itk::LinearInterpolateImageFunction`, 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.

How do I run the application?

The application takes one argument: the name of a parameter file. A valid parameter file contains 17 lines with the following format.

Line	**Parameter**	**Example**
1	Filename of the raw 3D fixed (target) volume.	`N:/ImageData/target.raw`
2	Endian-ness of fixed volume. Zero represents little endian and a non-zero number represents big endian.	`1`
3	Size of fixed volume in column-row-slice order	`256 256 26`
4	Fixed volume pixel spacing in column-row-slice order	`1.25 1.25 4.0`
5	Filename of the raw 3D moving (source) volume	`N:/ImageData/source.raw`
6	Endian-ness of moving image. Zero represents little endian and a non-zero number represents big endian.	`1`
7	Size of moving volume in column-row-slice-order	`512 512 29`
8	Moving volume spacing in column-row-slice order	`0.653595 0.653595 4.0`
9	Axes permutation order to bring the moving image in the same orientation as the fixed image. (E.g. 1 2 0 means that the	`1 2 0`

	column (0) axis of the fixed image corresponds to the row (1) axis of the moving image; the row (1) axis of the fixed corresponds to the slice (2) axis of the moving; the slice (2) axis of the fixed corresponds to the column (0) axis of the moving)	
10	Three numbers indicating which of the moving image axes (after permutation) needs to be flipped. Zero represents no flipping and a non zero represents flipping.	`0 1 1`
11	Number of multi-resolution levels to be used.	`5`
12	The starting (zeroth) level shrink factor for the fixed volume.	`4 4 1`
13	The starting (zeroth) level shrink factor for the moving volume.	`8 8 1`
14	The number of iterations to be performed at each resolution level.	`2500 2500 2500 2500 2500`
15	The learning rate at each resolution level.	`1e-4 1e-5 5e-6 1e-6 5e-7`
16	The scale applied to the translation parameters during optimization.	`320`
17	The output directory while the PGM files are to be written to. (NB. directory must already exists otherwise no images are written out)	`pgmsdir`

Both the fixed and moving volumes are assumed to be in binary (signed short) format. Differences in the patient orientation can be taken into account by specifying the permutation order and which axes which require flipping.

Lines 11 to 13 specify the multi-resolution down-sampling schedule. Line 11 specifies the number of multi-resolution levels to be used. Line 12 and 13 respectively specify the starting (zeroth) level shrink factor for the fixed volume and moving volume. The shrink factors for all other levels are computed automatically by dividing the shrink factors of the previous level by a 2. All shrink factors less than 1 is rounded to a value of 1.

Line 17 specifies the directory where the output PGM files are to be written. Slices from the fixed volume are prefixed `target` then followed by the slice number starting from `000`. Similarly, slices from the moving volume are prefixed `source` and slices from the registered image are prefixed `register`.

Some example experiments

The Vanderbilt Retrospective Registration Evaluation Project contains multi-modal images (CT, MR-T1, MR-T2, MR-PD, PET) of ten patients. Please refer to the information at:

`http://cswww.vuse.vanderbilt.edu/~image/registration/`

The following results were produced using parameter file `SampleInputs/PracCTToT1.txt`.

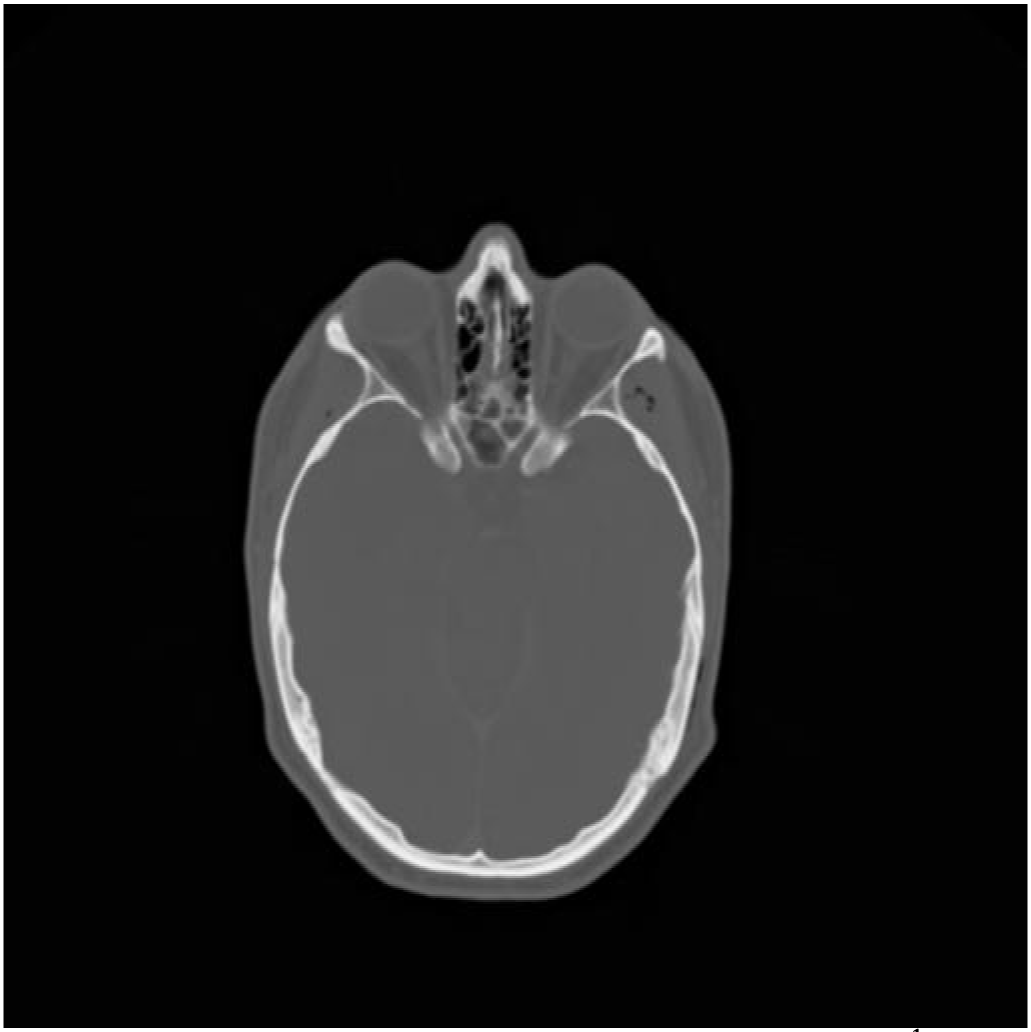

Fig 1: Slice 7 from the practice patient CT volume (the moving volume)[1]

[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.

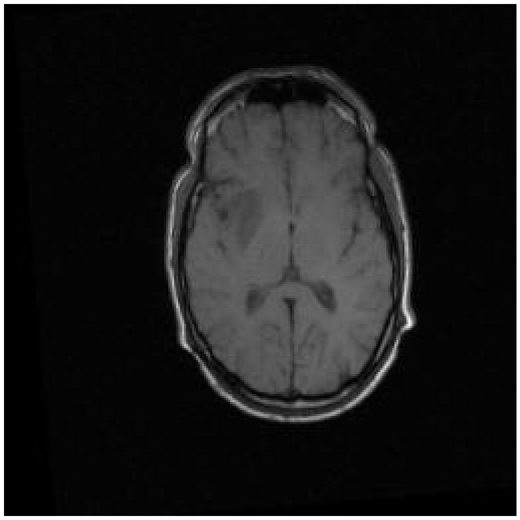

Fig 2: Slice 7 from the practice patient MR-T1 volume (the fixed volume)

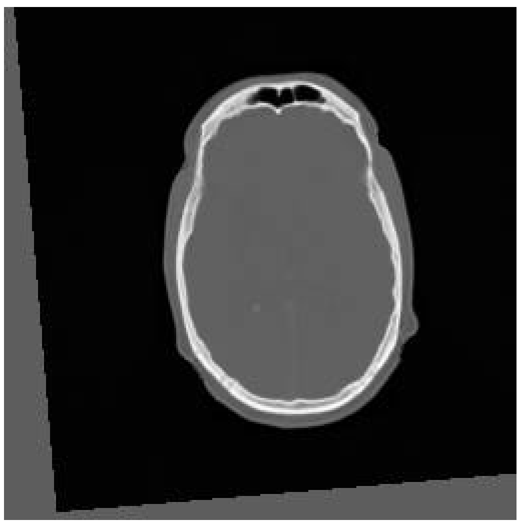

Fig 3: Slice 7 from the CT-to-MR-TI registered image

The following results were produced using parameter file `SampleInputs/PracPETToPD.txt`.

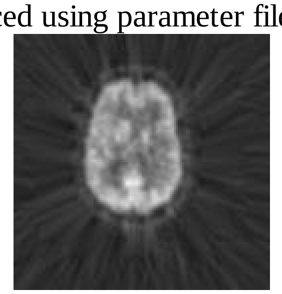

Fig 4: Slice 7 from the practice patient PET volume (the moving volume)

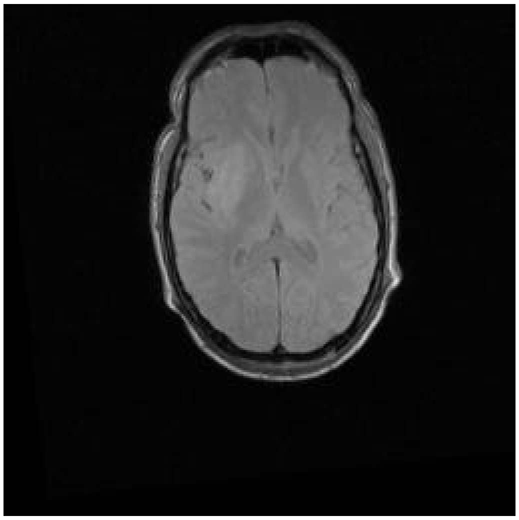

Fig 5: Slice 7 from the practice patient MR-PD volume (the fixed volume)

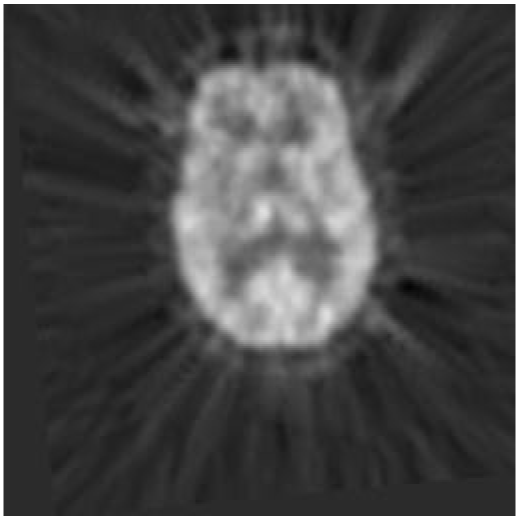

Fig 6: Slice 7 from the PET-to-MR-PD registered image

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.