

A development environment for medical image registration procedures

Matjaž Šuber
University of Primorska
Faculty of Mathematics, Natural Sciences
and Information Technologies
Glagoljaska 8, 6000 Koper
Koper, Slovenia
matjaz.suber@gmail.com

ABSTRACT

Development of novel image registration methods and procedures is a tedious task because of the need of large number of specialized lower level functions and an environment with the ability to observe results of each step of the processing. Although several software libraries and environments are available to simplify the development process, they require exhaustive learning and adaptation. To overcome this problem we aimed to implement a development environment that offers all this with only minimal influence on the user development strategies. We have decided to build it on top of Matlab environment that is already well established in this research field. The image registration environment includes a graphical user interface that simplifies testing and application of image registration procedures as well as their development by incorporating a large set of lower level processing functions. The environment was tested by using it for the development of the rigid registration procedure. The results clearly showed that we obtained a reliable environment, which has simplified and improved the implementation.

Keywords

Medical image registration, Development environment, Graphical user interface, Image registration toolbox

1. INTRODUCTION

Medical image registration is a very important component of many clinical applications. It can be used for the detection and diagnosis of diseases, for planning the therapy, for guidance of interventions and for the followup and monitoring of patients.

The primary goal of image registration is to find the corresponding anatomical or functional locations in two or more images. Image registration can be applied to images from

the same subject, acquired by the same (mono-modal image registration) or by different (multi-modal image registration) imaging modalities or at different time points (serial image registration).

In general, the process of image registration involves finding the optimal geometric transformation which maximizes the correspondences across images. Depending on the problem, geometric transformation can model only rotations, translations and scaling of images (rigid transformation) or can include deformations (nonrigid transformation).

In order to provide high quality image registrations, researchers need development tools, which simplify development and testing of methods by offering easy accessible supporting functions. Such an environment needs to work with 3D images, to provide a slice by slice visualization, to compare the geometric relations between images, to obtain image statistics like image histograms, to apply and display registration results and to provide tools that helps implementing new registration procedures.

Development of image registration methods can aid from different types of supporting software including programs like 3DSlicer [4] or ImageJ [1], programmer libraries like ITK [2] and VTK [10] or toolboxes for common development environments like Matlab [11] or Python [9]. The problem of most of these tools is that they require user adaptation and learning.

In this paper we describe a new image registration environment for Matlab, that improves the implementation of image registration procedures. This environment includes a graphical user interface that simplifies testing and application of image registration procedures as well as their development by incorporating a large set of lower level processing functions. Its advantage is that it is straightforward and does not need extensive learning. In the next section we will firstly describe our environment, in section 3 we show how it can support the development and testing of registration procedures on an example of developing a rigid registration procedure. In section 4 we conclude the paper with a discussion.

2. SPECIFICATIONS

The image registration environment is meant to improve and speed-up the process of implementing new image registration procedures. In order to improve and speed-up the implementation phase this environment needs to satisfy the following requirements.

The first requirement is a data structure that will store images and related registration properties, such as image origin, global transformation, region of interest and much more. This data structure is meant to help implementing different image registration procedures in a common way. The second requirement is a graphical user interface that simplifies work with medical images and registration techniques. It needs to work with multiple 3D medical images which are saved in different file formats. The third requirement is the possibility to select different image registration procedures and to display their results. The fourth requirement are various tools that can be easily used to implement image registration components, such as measuring similarity using different similarity measures, resampling images and much more.

2.1 Data structure

The first requirement of the registration environment is a well defined data structure that stores images, their properties, registration results and other registration parameters. The main reason of such data structure is to provide a common way to manipulate images in different implementations of registration procedures and to display those images and registration results in the graphical user interface. The described data structure unifies data requirements of previous image registration implementations and is organized as an array of images. It includes a predefined set of properties which can be easily extended to suffice the needs of image registration procedures.

Predefined properties of the data structure:

RefIdx: Index of the image in the `Img` array to be used as a reference image.

MovIdx: Index of the image in the `Img` array to be used as a moving image.

Img: Array which stores all medical images and related registration properties.

Img[i]: Structure defining a single medical image and related registration properties.

Img[i].name: Image name.

Img[i].path: Absolute path to the image data.

Img[i].voxelSize: Image voxel size in millimeters.

Img[i].data_orig: Original image data in case of losing image information in the process of conversion to `uint8` format.

Img[i].data: Image data used in registration procedures which is converted to `uint8` format. The conversion is an established compromise between the necessary image bright-

ness and the statistical significance of the registration procedure.

Img[i].roi: Image coordinates of region of interest.

Img[i].O: Image origin in millimeters. It is the point around which the global image registration transformations are defined.

Img[i].T: 3D global geometric transformation defined as 4x4 matrix.

Img[i].D: Deformation field that enables describing complex, including nonrigid geometric transformations, defined as a 3D displacements of individual image voxels.

2.2 Graphical user interface

The second requirement of the image registration environment is a graphical user interface that simplifies work with medical images and registration techniques. It allows loading 3D images in different file formats like DICOM [7], BrainWeb [3] and RIRE (Retrospective Image Registration Evaluation project) [12]. It provides a slice by slice visualization, image statistics like histograms and comparison of geometric relations between images. The graphical user interface also allows exporting and importing the described data structure in order to provide the capability to save temporary results that can be reused in future. One of the main features of the graphical user interface is to load and execute different image registration procedures. This feature simplifies and improves the testing phase of the development process.

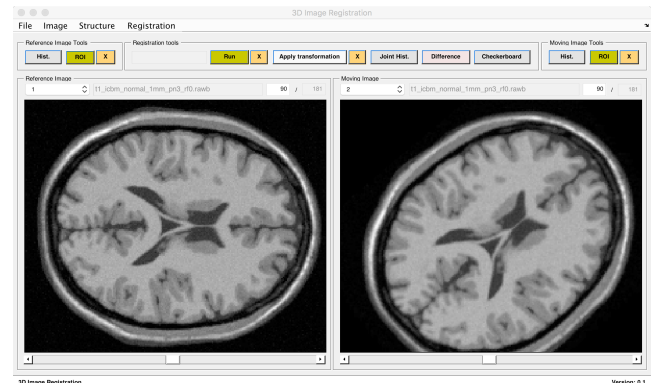


Figure 1: Graphical user interface.

The graphical user interface provides a set of tools that helps researchers to analyze image relations and registration results. This tools are divided into two groups. The first group consists of tools that can be executed over a single image like plotting a histogram as shown in figure 2 or setting a region of interest.

The second group consists of tools that helps analyzing geometric relations between two images. In this scope the graphical user interface includes displaying the joint intensity distribution histogram [8], absolute difference and checkerboard image.

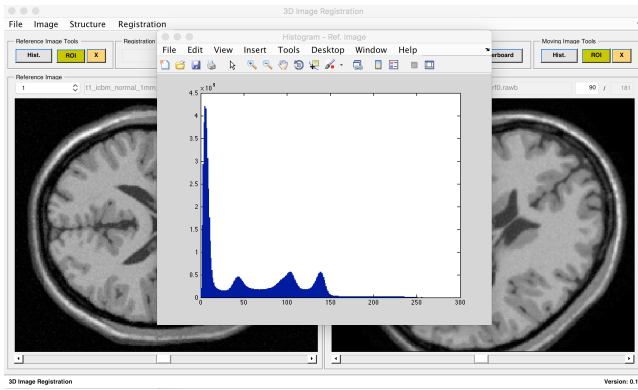


Figure 2: Image histogram as an example of the first set of tools.

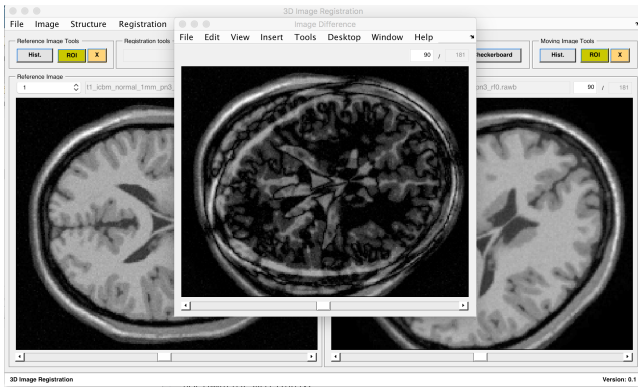


Figure 3: Absolute difference as an example of the second set of tools.

The graphical user interface is build with Matlab GUI [6] toolbox and can be easily initialized within Matlab workspace.

2.3 Registration Toolbox

Image registration procedures typically consists of three components. The first component is the geometric transformation which describes the movement, rotation, translation and/or deformation of objects in the image. The second component is the similarity matrix, which is used to estimate degree of image alignment. The third component is the optimization process. The goal of it, is to find the optimal geometric transformation which maximizes or minimizes the similarity of two images.

Implementation of these components may require a lot of effort if not supported by lower level processing functions. We have collected a large set of such lower level processing functions that were implemented in past research activities. This functions are grouped into a toolbox that can be used through the graphical user interface or directly in the Matlab workspace. This toolbox will be constantly updated according to the needs of future image registration procedures.

3. RESULTS

We have tested the image registration environment by using it for the development of a rigid registration procedure and

demonstrate it on BrainWeb images.

We started by creating a new project on the top of the registration environment. Without any other interference we initialized the graphical user interface wich is shown in figure 4.

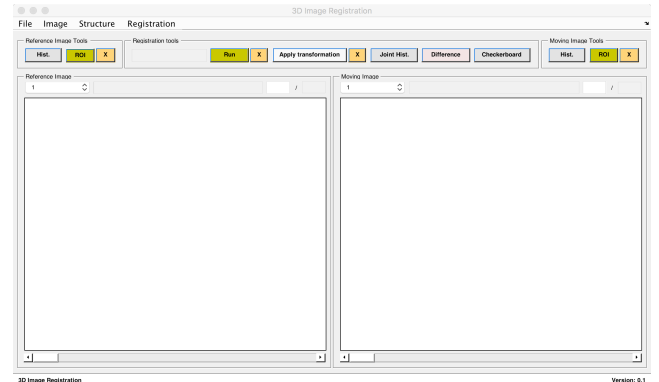


Figure 4: Graphical user interface.

After that we loaded the reference and moving image into the registration data structure through the user interface. For the reference image we used a MRI image of brain with $181 \times 217 \times 181$ voxels, 1mm slice thickness and with 3% of noise from the BrainWeb database. To create the moving image we applied a rigid rotation of 10 degrees over the reference image. The user interface displayed images as show in figure 1. Without any additional coding we could slide through all slices of the reference and moving image. In addition to that we analyzed the relation between the two images by displaying the joint intensity distribution histogram or the absolute difference as shown in figure 3.

At this point we only needed to focus on the core implementation of the rigid registration procedure. The step of creating the necessary environment was done in a very short time. To implement the rigid registration procedure we created three components. The first component was designed to controll the overall registration workflow. The second component was designed to compute the similarity between the reference and the moving image. This component was implemented by using low level processing functions from the integrated toolbox. The third component was the optimization process. It was implemented with the free/open-source NLopt [5] library which provides implementation of several optimization methods.

After we implemented the rigid registration procedure on the top of the registration data structure we started testing the implementation. In the graphical user interface we selected the implemented rigid registration method as shown in figure 5. We executed the procedure many times until we solved all issues and got the preferred results as shown in figure 6.

4. DISCUSSION AND CONCLUSION

From the described results we can conclude that the registration library has satisfied all the initial requirements. The predefined data structure helped to implement more structured registration methods and to easily display regis-

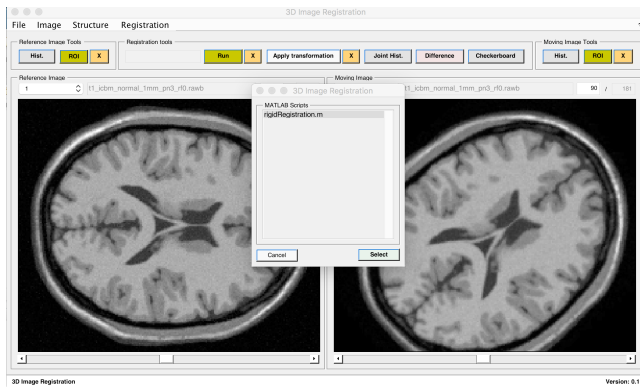


Figure 5: File selector for selecting image registration procedures.



Figure 6: Graphical user interface after image registration.

tration results in the graphical user interface. Without any additional coding the graphical user interface displayed the imported medical images and provided all necessary tools for analyzing them. The testing phase was reasonably easier because of the possibility to apply the registration results directly on the loaded moving image inside the user interface. The integrated toolbox helped to speed up the implementation phase of the registration procedure. In the end we only needed to focus on the implementation of three small components. The first one was to control the overall registration workflow, the second was to compute the similarity between images and the last component was the optimization process. To conclude, we shall summarize our contribution presented in the paper. We have implemented a new image registration environment for Matlab. It offers support for development of image registration procedures. This environment includes a graphical user interface that simplifies testing and application of image registration procedures as well as their development by incorporating a large set of lower level processing functions. To provide a common way for manipulating images and registration results it includes a well defined data structure. The main goal of this environment is to provide a comprehensive solution to simplify and to speed-up the development of image registration procedures. We have used this library in a new project in which we implemented a rigid registration procedure for medical images. From the described results we can conclude that

the development of such procedure within the new image registration environment was much faster and reliable.

5. ACKNOWLEDGMENT

We thank Peter Rogelj, professor at University of Primorska, for assistance in the implementation of image registration environment and for the provided set of low level processing functions that were implemented in past research activities.

6. REFERENCES

- [1] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram. Image processing with imageJ, 2004.
- [2] B. B. Avants, N. J. Tustison, M. Stauffer, G. Song, B. Wu, and J. C. Gee. The Insight ToolKit image registration framework. *Frontiers in Neuroinformatics*, 8, 2014.
- [3] C. Cocosco, V. Kollokian, R. K. Kwan, G. B. Pike, and A. C. Evans. BrainWeb : Online Interface to a 3D MRI Simulated Brain Database. *3-rd International Conference on Functional Mapping of the Human Brain*, 5(4):S425, 1997.
- [4] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J. C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, 30(9):1323–1341, 2012.
- [5] U. Kumar, S. Soman, and Jayadeva. Benchmarking NLOpt and state-of-the-art algorithms for continuous global optimization via IACOR. *Swarm and Evolutionary Computation*, 27:116–131, 2016.
- [6] C. S. Lent. Learning to Program with MATLAB Building GUI Tools. *John Wiley and Sons Inc.*, page 310, 2013.
- [7] P. Mildenerger, M. Eichelberg, and E. Martin. Introduction to the DICOM standard, 2002.
- [8] D. Mistry, A. Banerjee, and A. Tatu. Image Similarity based on Joint Entropy (Joint Histogram). *International Conference on Advances in Engineering and Technology*, (1):5, 2013.
- [9] G. V. Rossum, P. S. Foundation, U. Swallow, S. Python, P. Software, and F. License. Python (programming language). *Flying*, pages 1–14, 2011.
- [10] W. J. Schroeder and K. M. Martin. The visualization toolkit. In *Visualization Handbook*, pages 593–614. 2005.
- [11] The Mathworks Inc. MATLAB - MathWorks, 2016.
- [12] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer, R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. Hill, D. J. Hawkes, C. Studholme, J. B. Maintz, M. A. Viergever, G. Malandain, and R. P. Woods. Comparison and evaluation of retrospective intermodality image registration techniques. *International Society for Optics and Photonics*, pages 332–347, 1996.