Technische Universität München
Fakultät für Informatik
Prof. N. Navab, Ph.D.

SS 2013
Exercise Sheet 4
May 16, 2013

Exercises in Computer Aided Medical Procedures II

Today you will learn the intensity based registration and its main components, i.e. Similarity Measure, Transform and Optimizer as shown in Fig. 1. You will perform 2D-2D rigid registration. Therefore, for transformation you will first implement translation and rotation. Then, you will implement different cost functions(i.e. similarity measures). Finally, you will combine these components together with an optimizer to optimize for the best registration parameters.

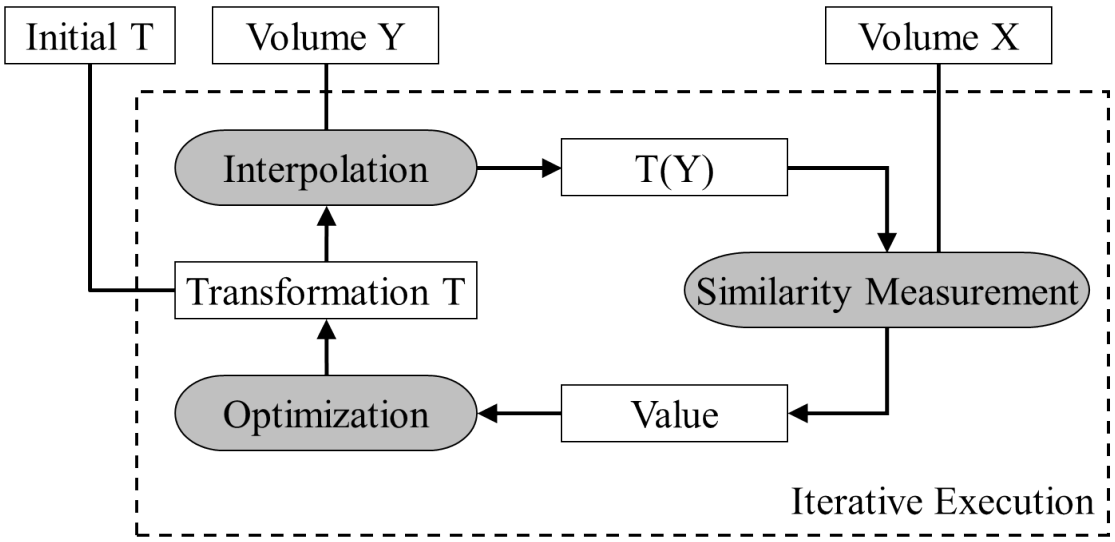

Figure 1: Main components of registration

Exercise 1 (P) MATLAB - Path

Include the directory *04IntensityBasedRegistration* into Matlab's path (File -> Set Path).

Exercise 2 (P) Transformation

Open `image_translate.m`. Here, you will implement the translation by filling in the code for computing the indices for image. `image_rotate.m` is given for you.

Exercise 3 (P) Optimization

Open `ibRegistration.m`. This file is a test script driving the intensity based registration. It loads the data, performs the initialization, calls the optimizer and displays the result of the registration. Do the followings:

- Go to `Optimizer`: You don't have to do anything here! Understand the call of the optimizer *fminsearch*.
- Go to `apply resulting transformation`: Fill in the transformation code to apply the final registration transform. *Hint*:Use `image_rotate.m` and `image_translate.m`.

Exercise 4 (P) Similarity Measure

Open `cost_function.m`. This file contains the evaluation of different similarity measures. Optimizer will call this function at each iteration to asses the similarity.

- Go to `apply current transformation`: Fill in the transformation code to apply the current transform on the moving image. *Hint*:Use `image_rotate.m` and `image_translate.m`.
- Go to `calculate SSD`: Fill in the calculations for the SSD similarity measure, test your code using the test script `ibRegistration.m`.
- Go to `calculate SAD`: Fill in the calculations for the SAD similarity measure, test your code using the test script `ibRegistration.m`.
- Go to `calculate NCC`: Fill in the calculations for the NCC similarity measure, test your code using the test script `ibRegistration.m`.
- Go to `calculate MI`: Fill in the calculations for the MI similarity measure. You have to first implement `joint_histogram.m` to calculate the joint histogram. Test your code using the test script `ibRegistration.m`.

Exercise 5 (P) Miscellaneous - Bonus!

- Create or find registration scenarios (i.e. images) for which the registration using:
  - SSD/SAD does not work, but using NCC it does work
  - SSD/SAD or NCC does not work, but using MI it does work
- Play around with different images and initial transforms in order to observe how the registration results are changing.
- Place everything in a multi-resolution setting. For that you need to first create a Gaussian image pyramid as shown in Fig. 2 by implementing a method `gaussian_pyramid.m` which returns a set of images at different levels. You have to share information between pyramid levels.
- Optionally, you can implement a bilinear interpolation in `image_translate.m` to obtain a better alignment.

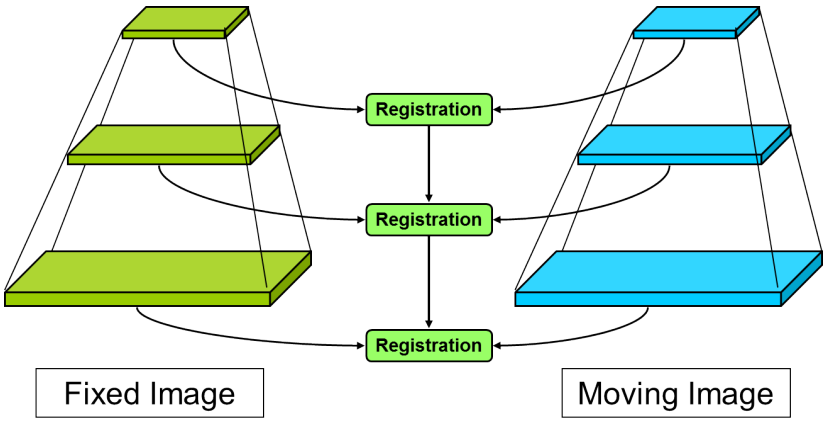

Figure 2: Multi-resolution image registration.