Project 2: Registration of different MRI modalities

TSAI

November 23rd, 2020

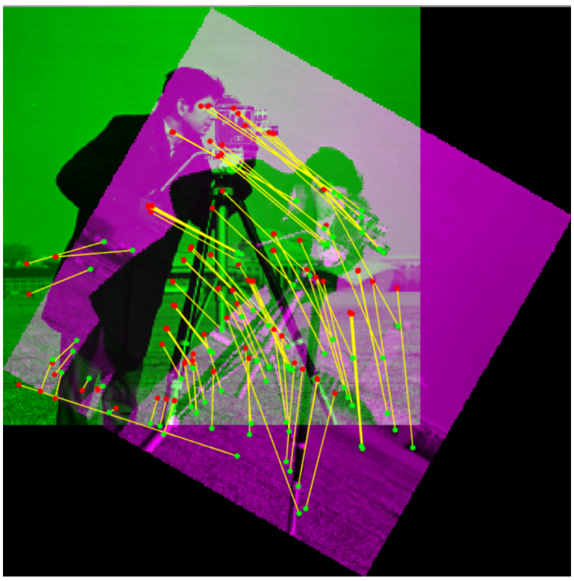
Referent:

- Noëlie Debs, noelie.debs@creatis.insa-lyon.fr

Introduction

Image registration consists in aligning two images that are initially shifted. It is one of the two major issues in the field of image processing, with segmentation.

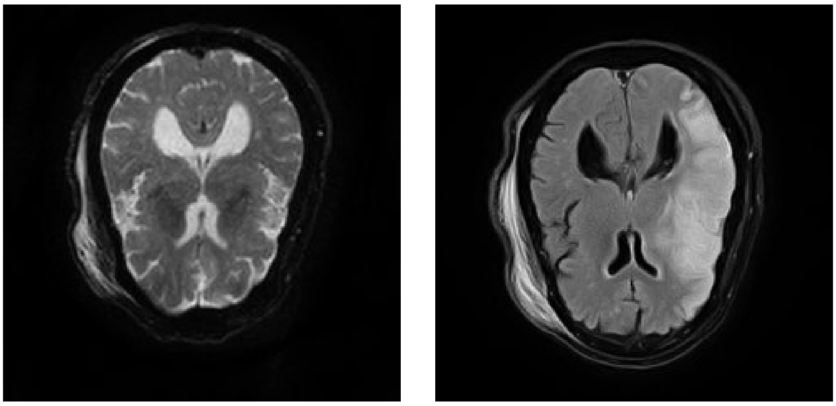
The mapping is done by searching for geometric transformations ρ (translations, rotation, *etc*) to move from one image to another. One of the images is considered as reference, it is the *fixed* image. We apply to the other, the *movinf* image, a succession of geometric transformations. At each transformation, the alignment between the fixed and mobile images is calculated according to a pixel-to-pixel comparison criterion previously defined. We then preserve the ρ transformation which allowed the best alignment: the moving image, once transformed, is called the *registrated* image.



Images at your disposal

In this project, you have at your disposal 2 images (Diffusion MRI and FLAIR MRI) from 3 patients with stroke. The MRI-FLAIR was systematically performed 6 days after the Diffusion MRI, which means that the patient do not have exactly the same position in the 2 images.

The objective of this project is to register the Diffusion MRI at best on the FLAIR MRI. Thus, the FLAIR image is the fixed image, and the Diffusion MRI is the moving image to be registered.



(a) Diffusion MRI

(b) FLAIR MRI

Figure 1: Slice z=8 from patient 1 shown for 2 modalities: diffusion MRI (left), FLAIR (left)

Notes

- For this project, you will need to create a program capable of automating the registration process.

1 Pre-processing

The pairs of images require a pre-processing before registration.

1. Note that the two images do not have the same size. Reduce the size of the largest image to the size of the smallest image (*pixel downsampling*).

2. We want to keep only the brain on the image, and zero all the rest of the image (in other words, we want to extract the cranial box and ventricles from the original images). Write a function able to realize that.
   Start by thresholding the original image, then recover the sizes of the created blob in the threshold image. The brain should be the biggest blob.
   Then create a "mask" image (*ie* a binary image which will be black everywhere except in the brain pixels it will be white). Use this mask to keep only the pixels of the brain in the original image.

2 Rigid registration

Rigid registration offers ρ transformations limited to t translations and r rotations. The FLAIR modality corresponds to the fixed image. The Diffusion modality is therefore the moving image. You will apply to it rotations and translations with different parameters, and calculate each time the similarity of this image with the fixed image.

1. First convert all images into grayscale images.

2. Start by calculating an initial estimate of the 2 necessary translations (a t_x translation on the x axis and a t_y translation on the y axis) between the two images. To do this, calculate the

center of gravity (centroid) of each image. Once both barycentres are calculated, calculate the 2 translations (t_x et t_y) between the 2 points. They will correspond to your initial translations t_{x0} et t_{y0}. The initial rotation r_0 will be a zero rotation.

3. Choose six values t_{min}, t_{max}, t_{step}, r_{min}, r_{max} and r_{step} that will be the lower and upper bounds as well as the step for the applied transformations. Be careful not to choose an interval too big to not explode in terms of computation time!

4. Implement the different transformations that you will apply to the moving image (a triple *for* loop that applies translations between t_{min} and t_{max} with a t_{step} step. Same for rotation).

5. For each transformation $\rho_i = (t_x, t_y, r)$, calculate the similarity criterion between the two images. Store the ρ_i value that gives the best similarity criterion: it will be our final transformation able to register the moving image. The similarity criterion will be as follows: $s = \sum_{i,j}(fixe[i,j] - mobile[i,j])^2$ This value corresponds to the (squared) difference between the intensity of each pixel from the fixed image and its corresponding pixel from the moving image.

6. Comment on your results. To guide you, you can answer the following questions: does the registration seem effective? Are there any cases where it fails? Can you understand why? What would you suggest to improve it?

3 Point set registration

A variant of the registration, which requires more interaction with the user but can give better results is the point set registration (or point matching): the user define recognizable points that he indicates in both modalities (by clicking on the specific pixels). The registration will then try to align these specific pixels with each other.

1. Implement the possibility for the user to enter specific pixels in each image modality.

2. Implement the registration which tries to align these specific pixels.

4 Non-rigid registration

Sometimes, rigid registration is not enough to register images correctly. Non-rigid registration allows non-linear transformations in addition to rotation and translation: *resizes*, homographies ("3D rotations"), découpes (*shear*), deformations..
The implementation is too complex to be done in this project, but it is implemented in other software like Matlab or Python.

1. Implement a non-rigid registration for your images (`imregister` function in Matlab. See tutorial: https://www.mathworks.com/help/images/ref/registration.metric.mattesmutualinformation.html

2. Comment on your results Does the non-rigid registration provide something for our application case?

5 Additional similarity metric

In section 2, try another similarity metric: mutual information. Unlike the least squares metric used in the previous question, it does not correspond to the difference between pixel value. The mutual information of two grayscale images measures how much an i value from a grayscale image 1 is correlated with another j value from a second grayscale image 2. For example, of $i = 180$ and $j = 212$, then we will say that these values are highly correlated if a large number of pixels are worth 180 in the image 1 and 212 in the image 2. The mutual image information calculates these values for each gray level combination ($i = 0...255$ and $j = 0...255$ for 8-bit images). Two images

will have a strong similarity if many i, j pairs have a strong correlation.
In other words, it is not necessary according to this criterion that two images have the same pixel values, it is enough that for a given pixel value i in image 1, the corresponding pixels in image 2 have for the most part the same value j (with i not necessarily equal to/nor close to j). This criterion is particularly useful for multi-modal images where structures correspond without having the same intensity.

To implement this, the formula for mutual information (see Wikipedia) is the following :
$\sum_{i=0}^{255} \sum_{j=0}^{255} p(i,j)log(\frac{p(i,j)}{p(i)*p(j))}$

Be careful, in this formula, i and j represent the different possible gray levels, not the pixels of the image.
$p(i,j)$ corresponds to the number of pixels with i value in the first image and j in the second.
$p(i)$ corresponds to the number of pixels that has the i value in the first image.
Implement this metric in 2.

Interesting to see this website (with a similar case to the one studied in this project) :
https://matthew-brett.github.io/teaching/mutual_information.html