Introduction to medical image registration
BIOMED

Pietro Gori

10 October 2018

This TP is about image registration using *MATLAB*. You will take two pictures of your hand and you will use them to test the algorithms presented during the lecture.

Data

You will create your own data ! Use the white paper on the table (or your own, it should be 21 cm of width and 22,5 cm of height), put your **right** hand wide open on it and take **at least two pictures** with your mobile phone (you can use mine if you do not have one) of your hand in different position. Here you have an example:

Figure 1: Two examples of picture of a wide open hand in different positions

Once taken the pictures, cut them using a software like Gimp to only have the white paper and your hand. After that, use the *Matlab* function *CreateData.m* downloaded with this TP to transform them in gray scale, resample at a given resolution ([570 520]) and select 22 landmarks. **IMPORTANT**: Select the landmarks in the same order and in the same way as in Fig.2.

1 Linear Registration

In the lecture, we define an affine registration as: $\mathbf{T}(x) = Ax + t$ where $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ and $t = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$ or, using homogeneous coordinates, $\mathbf{T}(x) = \begin{bmatrix} a_1 & a_2 & t_1 \\ a_3 & a_4 & t_2 \\ 0 & 0 & 1 \end{bmatrix}$. In *Matlab*, we use a

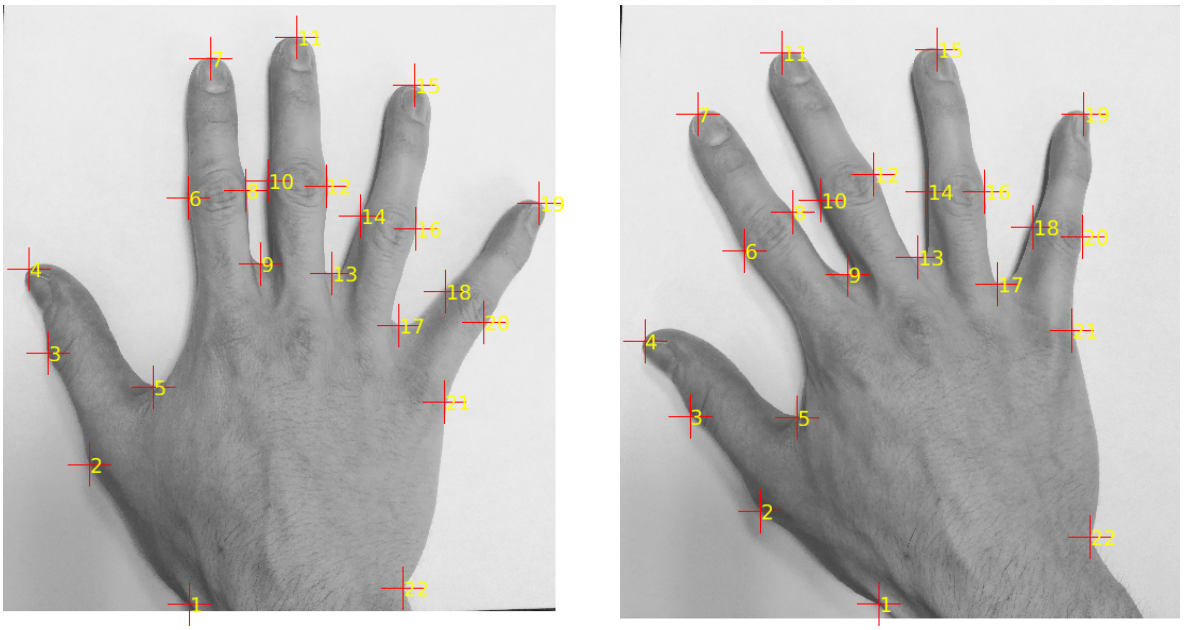

Figure 2: Example of landmark positioning

different notation: $\mathbf{T}(x) = \begin{bmatrix} a_1 & a_3 & 0 \\ a_2 & a_4 & 0 \\ t_1 & t_2 & 1 \end{bmatrix} = \begin{bmatrix} A^T & 0 \\ t^T & 1 \end{bmatrix}$.

1.1 Simple geometric transformations

Using the *Matlab* function *maketform('affine',A)*, or *affine2d* in the new Matlab versions, create a transformation and use it to deform the moving image A with the function B = *imwarp(A,tform)*. Use then the functions *imshowpair* and *imshow* to visualize the results. Create different transformations (i.e. scaling, rotation, shear, translation and combination of them) and apply them on your data.

You can of course also create your own functions. Something like:

```
function [T] = translate( tx, ty )
        T= [ 0 0 0; 0 0 0; tx ty 1 ];
end
```

Your T matrix can then be transformed into a Matlab transformation with the functions *Tm = affine2d(T)* (or maketform).

1.2 Intensity-based registration

Here, we will test an intensity-based registration technique. You can use the *Matlab* function *[optimizer, metric] = imregconfig('multimodal' or 'munimodal')* to define which intensity-based similarity measure , SSD or mutual information, and optimization scheme, gradient descent or OnePlusOneEvolutionary, you want to use. You can then use the *Matlab* functions *tform = imregtform(M, T, 'affine', optimizer, metric);* and *Mr = imwarp(M,tform,'OutputView',imref2d(size(T)));* to estimate the transformation. It is important to use the option *'OutputView',imref2d(size(T))* in order to visualize the deformed image in the reference frame of the target image !

1.3 Landmark-based affine registration

In this section, you will use the landmarks you previously positioned to estimate an affine registration. Implement an affine deformation using homogeneous coordinates (slide 39). Compare the result with the in-built function of Matlab *tform = fitgeotrans(Ml,Tl,'affine');*. Visualize the results using *Mr = imwarp(M,tform,'OutputView',imref2d(size(T)));* and *imshowpair(T,Mr)*.

```
1
2  function [T] = affine_landmarks(Ml,Tl)
3
4      [N,d]=size(Ml);
5      if size(Tl,1)~= N || size(Tl,2)~= d
6          error('size of two sets of landmarks should be equal')
7      end
8
9      T=zeros(3,3);
10     T(3,3)=1;
11
12         ....
13
14 end
```

1.4 Questions

1. Which is the best method ?
2. What happens when the two images you want to register are quite similar (for instance a slight rotation of 5 degrees) ? And when the two images are very different (translation and rotation of 90 degrees) ? You can take another picture of your hand or modify the one you have already taken.
3. What happens if you use less landmarks ? Try to remove some landmarks, does the result change ? Try different configurations and comment the results obtained.

1.5 Optional: non-linear registration

You can implement the landmark-based non linear registration seen this morning during the lecture (slide 44) or you can use the polynomial registration implemented in the *Matlab* function *tform = fitgeotrans(Ml,Tl,'polynomial',degree);*.

1. Try different degrees between 2 and 4 ? Do you obtain a better alignment ? What about the areas in the image far away from the landmarks when you use a degree equal to 4 ? Comment the results.