Image Analysis
Exercise 10 - Registration

2019

Introduction

The topic of this exercise is landmark based registration. In the first part, the goal is to align two photographs of hands and the second part is focused on examining the shape and size of a brain tumor before and after treatment.

As usual start by downloading the exercise data and start by creating a new m-file to contain the exercise scripts.

Part 1: Registration of Hand Photographs

The two hands are stored as `Hand1.jpg` and `Hand2.jpg`. Start by reading the two image files and show them using for example `subplot`.

```
hand1 = imread('Hand1.jpg');
hand2 = imread('Hand2.jpg');
```

We want to transform `hand 2` so it fits `hand 1`. Since `hand 1` is not moved it is the *reference image* and `hand 2` is the *template image*. In Matlab another notation is used. Matlab calls the reference image for the *fixed image* and the template image for the *moving image*. See the documentation for `cpselect`.

	Book	**Matlab**
Not changed	Reference image	Fixed Image
Changed	Template image	Moving Image

Landmarking

A landmark based method should be used to align the two images. The landmarking is done using the Matlab tool `cpselect`, where corresponding landmarks can be placed on the two images. Correspondence means that landmarks are placed on corresponding places in the two images. An example of landmarks placed on one hand can be seen in Figure 1.

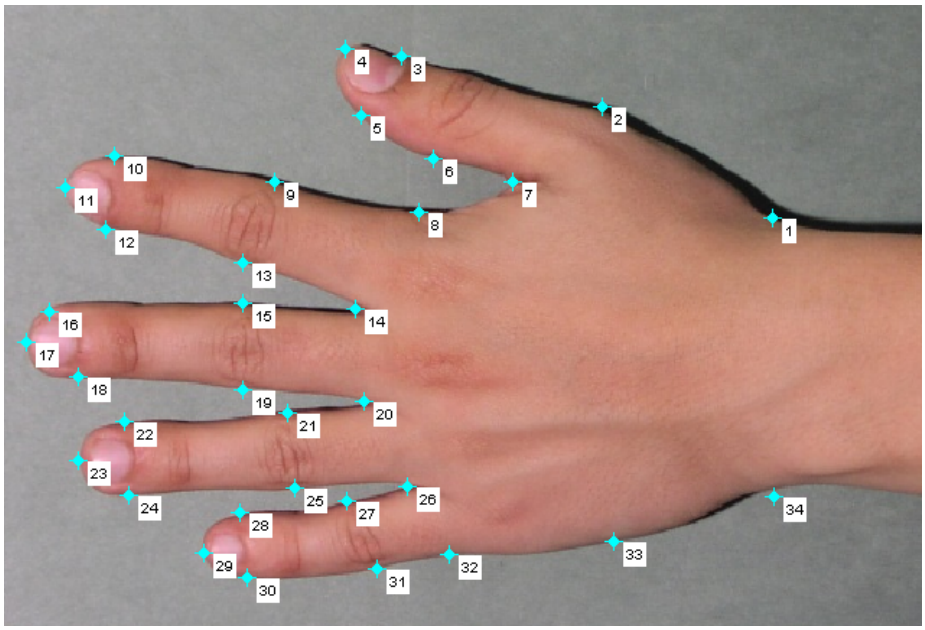

Figure 1: Landmarks placed on a hand.

Exercise 1 *Start* `cpselect` *with* ***`hand1`*** *as the fixed image and* ***`hand2`*** *as the moving image:*

```
cpselect(hand2, hand1);
```

Place corresponding landmarks around the hand in both images. You can choose the number of landmarks, but 10-15 should be suitable. Do this by first clicking for a landmark in the left window and then click on the corresponding point on the right window.

When you are finished finding landmarks you select the menu item ***`File | Export points to workspace`****. Then you can close* ***`cpselect`****.*

The found landmarks are now stored in two variables called `fixedPoints` and `movingPoints`. These can be saved to files on the disk, so they can be used later:

```
save('fixedPoints.mat','fixedPoints');
save('movingPoints.mat','movingPoints');
```

Exercise 2 *To visualise the annotation they can be plotted:*

```
figure;
plot(fixedPoints(:,1), fixedPoints(:,2), 'b*-', ...
     movingPoints(:,1), movingPoints(:,2), 'r*-');
legend('Hand 1 - The fixed image', 'Hand 2 - The moving image');
axis ij; % This reverses the direction of the axis
```

To calculate how well the two images are aligned from the start the objective function can be computed:

$$F = \sum_{i=1}^{N} \|a_i - b_i\|^2 \ , \tag{1}$$

here a_i is the coordinate of point i int the fixed image and b_i is the corresponding coordinate in the moving image.

Exercise 3 *Calculate F from your annotations. It can for example be done like this:*

```
ex = fixedPoints(:,1) - movingPoints(:,1);
errorX = ex' * ex;
ey = fixedPoints(:,2) - movingPoints(:,2);
errorY = ey' * ey;
F = errorX + errorY;
```

what is F?

The first transformation we are trying is a simple translation. It can be shown that a good estimate is to align the centre-of-mass of the two sets of points. The centre of mass can be computed as the average of the landmarks.

Exercise 4 *Compute the centre of mass of the two point sets. Store them in two variables called* `fixed_COM` *and* `moving_COM`*.*

To align the two set of landmarks the centre-of-mass can be subtracted from both.

Exercise 5 *Create two translated landmark sets by subtracting the computed centre-of-masses:*

```
fixed_trans = [fixedPoints(:,1) - fixed_COM(1) ...
    fixedPoints(:,2) - fixed_COM(2)];
moving_trans = [movingPoints(:,1) - moving_COM(1) ...
    movingPoints(:,2) - moving_COM(2)];
```

Plot the two translated landmark sets in the same plot. What do you observe? Do the hands match better than before?

The next transformation we are going to apply is the *similarity transform* that consists of a scaling, rotation and translation. In Matlab this transformation is called a *nonreflective similarity* transformation and it can be computed using `fitgeotrans`.

Exercise 6 *Compute the transform that makes hand2 fit hand1:*

```
mytform = fitgeotrans(movingPoints, fixedPoints,'NonreflectiveSimilarity');
```

The computed transform (a variable) `mytform` contains the rotation and translation necessary to transform points from hand2 to hand1. This can be verified using the function `transformPointsForward` that computes the transformed points.

Exercise 7 *Compute the transformed points from hand2 using:*

```
forward  = transformPointsForward(mytform,movingPoints);
```

plot them together with the points from hand1. What do you observe?

Exercise 8 *Use Equation (1) to calculate F after alignment. Do this by substituting b_i so they are now the transformed coordinates (forward). What has happened to the value of F. Can you explain it?*

Finally, the function `imwarp` can be used to transform an entire image using a computed transform.

Exercise 9 *Transform the image of hand2 using:*

```
hand2t = imwarp(hand2, mytform);
```

Show the transformed version of hand2 together with hand1. What do you observe?

Part 2: Analysis of a Glioma

The goal of this exercise is to examine and compare the shape and size of a tumor before and after treatment.

Exercise 10 *Find out what a glioma is and how it is typically treated.*

We will work with two magnetic resonance (MR) images of a giloma. `BT1.png` is taken before treatment and `BT2.png` is taken three months after treatment started. We now want to evaluate the results of the treatment by comparing the two images.

The position of the brain might not be the same in the two images, so the first thing we need to do is to make a transformation that compensate for this.

Exercise 11 *Use* `cpselect` *with BT1 as the fixed image and BT2 as the moving image. Choose at least 4 anatomical landmarks and place them around the brain (not the tumor). Save them to* `BT1points.mat` *and* `BT2points.mat`.

Exercise 12 *Use* `fitgeotrans` *and* `imwarp` *to create a similarity transformed version of BT2. The transformed image should keep the exact same size as the fixed image. This can be done by:*

```
TI = imwarp(I, mytform, 'OutputView', imref2d(size(I)));
```

Show the image of BT1 together with transformed version of BT2. What do you observe?

To be able to compare the tumor before and after treatment we start by segmenting it from the images. This could potentially be done automatically using pixel classification, but this is probably quite difficult. Instead we use manual delineation of the tumor.

Exercise 13 *Use* `roipoly` *to draw precisely around the edge of the tumor in BT1 and the transformed version of BT2. Store the two binary images as* `BT1_annot.png` *and* `BT2_annot.png`*. Show the two binary tumor images side by side and comment.*

We now want to examine how similar the two tumors are. A simple way to do this analysis is to add the two binary images together.

Exercise 14 *Add the two binary images together (BT1 and the transformed version of BT2). Use* `label2RGB` *to visualise the differences. Try to add two times the second image and use* `label2RGB` *again. Explain the output image.*

We now have an image that visualize how the tumor has changes during the three months of treatment.

Exercise 15 *Calculate how much of the tumor that was removed during treatment. You can do this by comparing the cross sectional area of the tumor before and after.*