Medical Image Analysis - Point based registration exercise

Mikkel Damgaard Olsen and Olivier Jais-Nielsen

September 23, 2009

1 Fiducial points

1.1 Selection of the fiducial points

We choose 10 points from the color cryosection corresponding to features that are precisely displayed in the different images. Cf. figure 1

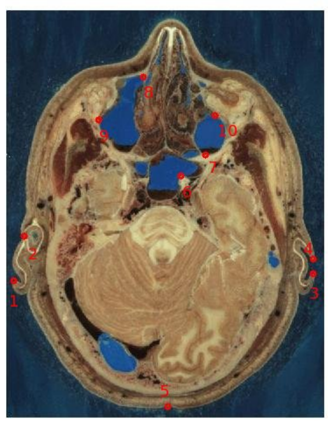

Figure 1: The control points on the cryosection

1.2 Fiducial Localization Error

For each of the fiducial points, both of us annotate the coresponding coordinates on the cryosection. For each point, we compute the average coordinates, which we will use as control points. We also compute the fiducial localization error (FLE) of each point.

The mean squared FLE corresponding to these values is approximately 1.0 pixel. This means the average error is of 1 pixel wich is negligible in comparison to the image dimensions.

According to the FLE criterium, the chosen fiducial points are satisfactory; thus we annotate their coordinates in the other images.

2 Transformation between MR and CT scans

2.1 Appropriate transformation

If we compare the positions of the features that are present in both images, for instance the bones, we can see that there is no significative non-rigid transformation nor distortion between them. The asymetries of each image are respected (for instance, the small white dot at the bottom of the skull is slightly on the right in both images) thus there is no reflection between the images. However, the scale is different in each image. Therefor, we consider the transfromation between the images to be an Euclidian similarity. Cf. figure 2

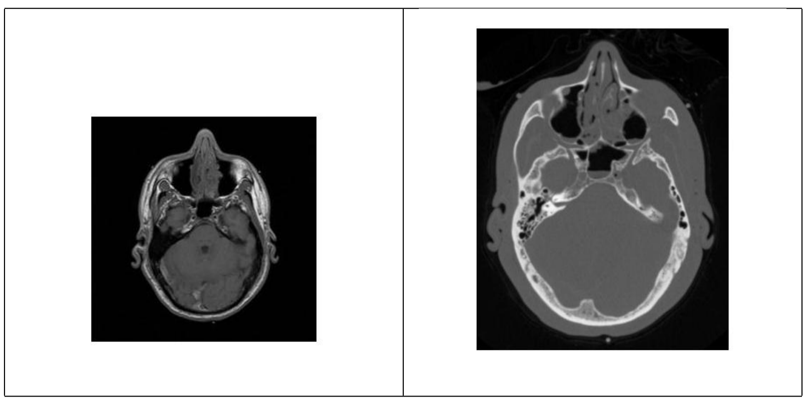

Figure 2: Left : MR scan. Right : CT scan.

2.2 Evaluation of the parameters

We can decompose an Euclidian similarity (T) according to the following formula : $T(x) = s.R_\theta.x + t$ where s is the positive scale, R_θ is the rotation of angle θ and t is a translation vector.

Matlab can compute T from the fiducial points.

A value of t is given by applying the transformation to 0, according to the formula $T(0) = t$. We find $t \approx \begin{pmatrix} -67 \\ -21 \end{pmatrix}$.

We consider the first vector e of the coordinates base. Let $u = T(e) - t$. We have $T(e) - t = s.R_\theta.e = s.\begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$ thus $\arctan\left(\frac{u_y}{u_x}\right) = \theta$ and $\|u\| = s$. We find $\theta \approx 1.7°$ and $s \approx 1.9$.

2.3 Evaluation of the parameters for the inverse transformation

We use the same method as above. The results are coherent with the formulas:

- Translation: $\begin{pmatrix} 35 \\ 9.8 \end{pmatrix}$ ($-\frac{1}{s}R_\theta^{-1}.t \approx \begin{pmatrix} 35 \\ 9.8 \end{pmatrix}$)
- Angle: $-1.7°$ ($-\theta \approx -1.7°$)
- Scale: 0.52 ($\frac{1}{s} \approx 0.53$)

2.4 Fiducial registration errors

For each fiducial point in the MR image, we compute the transformation of its coordinates and we compare this result to the coordinates manually annotated. We find a mean square FRE approximately equal to 20 pixels. This result remains satisfactory, in comparison to the image dimensions.

We use the same method for the inverse transform and we find a mean square FRE of approximately 5.4 pixels. This difference of values can be explained by the fact that as the inverse transform involves a downscaling, the required precision is lower: the distances are reduced therefor the errors are reduced as well. Indeed $s^2 \times 5.4 \approx 20$.

3 Result

After appliying the estimated transformation to the MR image, we crop both images so that they have the same size and create a colored image which red layer is the CT image, green layer is the transformed MR image and which has no blue layer. Cf. figure 3

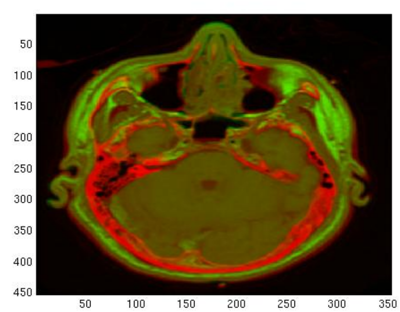

Figure 3: MR image and CT image superimposed

4 Matlab code

The source code can work in two ways:

- it only uses the images as inputs (`visiblehuman.mat`) and prompts the user for control points
- it uses the images as inputs as well as the fiducial points (Cf. table 1)

`average.ext`		`xyfresh.ext`	
% Average coordinates of the		% Coordinates of the fiducial	
% fiducial points in		% points in the "fresh" CT scan	
% the cryosection			
1.3407288e+01	4.6604126e+02	1.9674121e+01	2.1377476e+02
3.1413719e+01	3.8913880e+02	2.7233227e+01	1.7597923e+02
5.2621543e+02	4.5366184e+02	2.6156550e+02	2.0513578e+02
5.2734084e+02	4.2777760e+02	2.6264537e+02	1.8785783e+02
2.7750161e+02	6.8211844e+02	1.4385942e+02	3.1312300e+02
3.0075991e+02	2.8447642e+02	1.5465815e+02	1.2522524e+02
3.4164952e+02	2.4846356e+02	1.7409585e+02	1.0470767e+02
2.3586174e+02	1.1566613e+02	1.2226198e+02	4.0995208e+01
1.5933441e+02	1.8881726e+02	8.8785942e+01	7.7710863e+01
3.5890568e+02	1.8168971e+02	1.7841534e+02	7.4471246e+01

`xyfrozen.ext`		`xymri.ext`	
% Coordinates of the fiducial		% Coordinates of the fiducial	
% points in the "frozen" CT scan		% points in the MR scan	
1.6340256e+01	3.0380831e+02	4.7937700e+01	1.6775879e+02
2.9308307e+01	2.5337700e+02	5.4480831e+01	1.4322204e+02
3.3189617e+02	2.9084026e+02	2.1396965e+02	1.6039776e+02
3.3333706e+02	2.7066773e+02	2.1151597e+02	1.4240415e+02
1.8060224e+02	4.3060703e+02	1.3708786e+02	2.3319010e+02
1.9357029e+02	1.8709585e+02	1.3954153e+02	1.0396326e+02
2.1374281e+02	1.5683706e+02	1.4935623e+02	9.1694888e+01
1.4890256e+02	8.0469649e+01	1.1582268e+02	4.9982428e+01
1.0711661e+02	1.2946006e+02	9.2921725e+01	7.7790735e+01
2.2526997e+02	1.2225559e+02	1.5753514e+02	6.7158147e+01

Table 1: Files containing the fiducial points coordinates

The following source code implements the second of these two ways (automatic). However, to try the first way (manual), one has to comment out the lines that load the coordinates files and to uncomment the initially commented lines:

```
%% Loading of the images

load visiblehuman.mat;

```

```
%% Acquisition of the fiducial points for the cryosection (and computation of the FLE)

% cpselect(head,head)
% average=(input_points+base_points)/2;
% save ('average.ext', 'average', '-ASCII');
load average.ext;
nPoints=size(average,1);
% FLE=base_points-average;
% FLE_asl=(norm(FLE)^2)/nPoints

%% Display of the fiducial points as red circles

imshow(head);
hold on;
for k=1:nPoints
plot(average(k,1), average(k,2), 'or', 'linewidth', 3);
end

%% Acquisition of the fiducial points for the "fresh" CT scan

% figure
% imagesc(head_fresh)
% axis image
% colormap gray
% [x y]=ginput(nPoints);
% xyfresh=[x y];
% save('xyfresh.ext', 'xyfresh', '-ASCII');
load xyfresh.ext;

%% Acquisition of the fiducial points for the "frozen" CT scan

% figure
% imagesc(head_frozen)
% axis image
% colormap gray
% [x y]=ginput(nPoints);
% xyfrozen=[x y];
% save('xyfrozen.ext', 'xyfrozen', '-ASCII');
load xyfrozen.ext;

%% Acquisition of the fiducial points for the MR scan

% figure
```

```
% imagesc(head_mri)
% axis image
% colormap gray
% [x y]=ginput(nPoints);
% xymri=[x y];
% save('xymri.ext', 'xymri', '-ASCII');
load xymri.ext;


%% Computation of the transformation parameters for the MR --> "frozen" CT transformation

MR2CTtform=cp2tform(xymri,xyfrozen,'nonreflective similarity');
t_MR2CTtform=tformfwd(MR2CTtform,[0,0])
u=tformfwd(MR2CTtform, [1,0]) - t_MR2CTtform;
angle = (180/pi) * atan2(u(2), u(1))
scale = norm(u)


%% Computation of the FRE for the MR --> "frozen" CT transformation

[x_MR2CT y_MR2CT]=tformfwd(MR2CTtform,xymri);
FRE_MR2CT=[x_MR2CT y_MR2CT]-xyfrozen;
FRE_MR2CT_asl=(norm(FRE_MR2CT)^2)/nPoints

%% Computation of the transformation parameters for the "frozen" CT --> MR transformation

CT2MRtform=cp2tform(xyfrozen,xymri,'nonreflective similarity');
t_CT2MRtform=tformfwd(CT2MRtform,[0,0])
u=tformfwd(CT2MRtform, [1,0]) - t_CT2MRtform;
angle = (180/pi) * atan2(u(2), u(1))
scale = norm(u)

%% Computation of the FRE for the "frozen" CT --> MR transformation

[x_CT2MR y_CT2MR]=tformfwd(CT2MRtform,xyfrozen);
FRE_CT2MR=[x_CT2MR y_CT2MR]-xymri;
FRE_CT2MR_asl=(norm(FRE_CT2MR)^2)/nPoints

%% Creation of the superimposed image

```

```
[im_MR2CT xRange yRange]=imtransform(head_mri,MR2CTtform);
im1_Trans=im_MR2CT;
im2=head_frozen;

overLap(1:2)=max(ceil([[xRange(1) yRange(1)];[1 1]]));
overLap(3:4)=min(floor([[xRange(2) yRange(2)];[size(im2,2) size(im2,1)]]))-overLap(1:2);

im1_T_cropped=imcrop(xRange,yRange,im1_Trans,overLap);
im2_cropped=imcrop(im2,overLap);

im_RGB(:,:,1)=im2_cropped;
im_RGB(:,:,2)=im1_T_cropped;
im_RGB(:,:,3)=zeros(size(im2_cropped));
figure,imagesc(im_RGB);
```