**Report for Assignment 2 – Part II: Segmentation by Deformable Models**

**Ravikiran Janardhana**

**Medical Image Analysis – COMP 775**

**Fall 2011**

**Tasks completed**

* Wrote a Matlab program which reads n tuples of N/2 (xi,yi) points, centers each tuple by subtracting its center of gravity, computes the mean of the results, and translationally aligns each centered tuple to that mean.
* Plotted the mean face with the translated version of one of the sample faces superimposed on it.
* Computed the mean subtracted set and applied Principal Component Analysis using princomp function in Matlab.
* Recorded the first 3 eigenmodes and the associated principal variances (eigenvalues).
* Computed the face which is 2 standard deviations away from the 1st eigen vector, i.e., (µ + 2σ1v1) and plotted it superimposed on the mean face.

**Code**

% Read all the points

% pts contains all the 21 point sets (21x68x2)

% (:,:,1) - contains x values

% (:,:,2) - contains y values

files=dir('./dat/\*.pts');

for i = 1: length(files)

fName = strcat('./dat/',files(i).name);

pts(i,:,:) = readPoints(fName);

end

% backup the original points read

orig\_pts = pts;

% Compute Center of Gravity (Mean) of each point set and subtract each one

% of the points from the center of gravity of the same set

for i = 1 : size(pts, 1)

cg\_pts = squeeze(mean(pts(i,:,:)));

for j = 1 : size(pts, 2)

pts(i,j,1) = pts(i,j,1) - cg\_pts(1);

pts(i,j,2) = pts(i,j,2) - cg\_pts(2);

end

end

% Generate the mean for all the sets of points and draw it

mean\_of\_sets = squeeze(mean(pts));

drawFaceParts(mean\_of\_sets, 'g-');

% Overlap it with the translated set of original points

colors = {'r-', 'k-', 'y-', 'c-'};

for i = 1 : 1

drawFaceParts(squeeze(pts(i,:,:)), colors{i});

end

%Subtract the mean\_of\_sets from each of the point sets

for i = 1 : size(pts, 1)

for j = 1 : size(pts, 2)

pts(i,j,1) = pts(i,j,1) - mean\_of\_sets(j,1);

pts(i,j,2) = pts(i,j,2) - mean\_of\_sets(j,2);

end

end

% We have 21 sets of points and 68 points are present in each set ,

% in order to do principal component analysis, we need 21 x 136 (68x2)

% matrix, where, the matrix is as below:-

% 1 - [x1 y1 x2 y2 ..... x68 y68]

% 2 - [x1 y1 x2 y2 ..... x68 y68]

% ...............................

% 21- [x1 y1 x2 y2 ..... x68 y68]

mean\_sub\_set = zeros(size(pts,1), size(pts,2) \* size(pts,3));

for i = 1 : size(pts, 1)

for j = 1 : size(pts, 2)

% eg:- mean\_sub\_set(1,1) = pts(1,1,1) and mean\_sub\_set(1,2) = pts(1,1,2);

mean\_sub\_set(i,2\*j-1) = pts(i,j,1);

mean\_sub\_set(i,2\*j) = pts(i,j,2);

end

end

% Perform Principal Component Analysis on the above mean\_subtracted set

% COEFF - A p-by-p matrix, each column containing coefficients for one

% principal component. The columns are in order of decreasing component

% variance

% SCORE - the principal component scores

% latent - a vector containing the eigenvalues of the covariance matrix of X

[COEFF,SCORE,latent] = princomp(mean\_sub\_set);

% Extract the first three eigen values and eigen modes of variation

eigen\_values = latent(1:3);

eigen\_modes = COEFF(:,1:3);

% Compute the 2 Standard deviations away face from 1st Eigen Vector

two\_sigma\_face = zeros(size(mean\_of\_sets,1), 2);

for i = 1 : size(mean\_of\_sets)

two\_sigma\_face(i,1) = mean\_of\_sets(i,1) + 2 \* sqrt(eigen\_values(1)) \* eigen\_modes(2\*i-1,1);

two\_sigma\_face(i,2) = mean\_of\_sets(i,2) + 2 \* sqrt(eigen\_values(1)) \* eigen\_modes(2\*i,1);

end

% Plot the mean face and the face which is two standard deviations away

% from 1st eigen vector

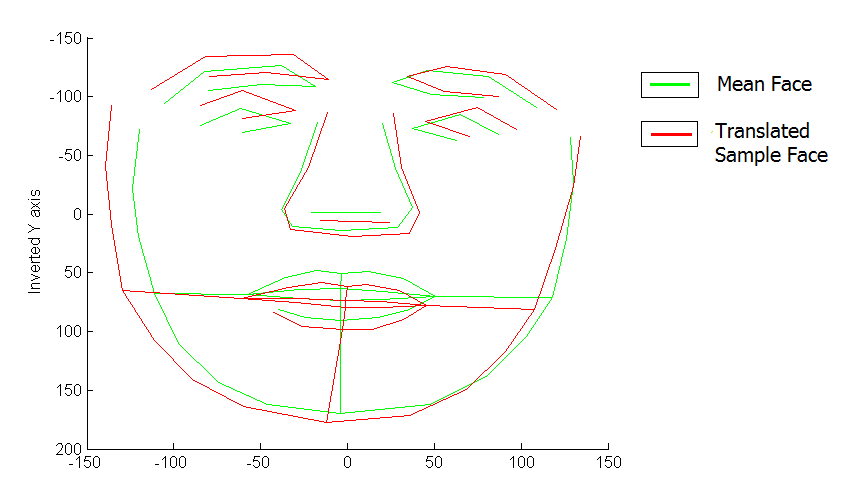
figure;

drawFaceParts(mean\_of\_sets, 'g-');

drawFaceParts(two\_sigma\_face, 'r-');

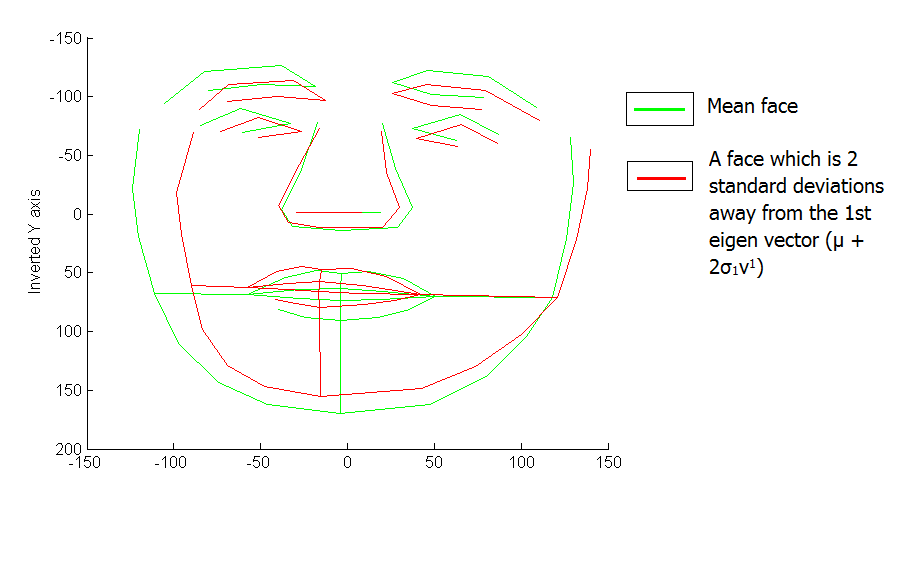
**Output**

**Figure 1: Mean face and a superimposed translated sample face.**

****

(Note: The Y axis has been inverted to show the face properly else the image would be upside down)

**Figure 2: Mean face and a superimposed face which is 2 standard deviations away from the first eigen vector.**

****

(Note: The Y axis has been inverted to show the face properly else the image would be upside down)