Comp 775, Medical Image Analysis — Fall 2011, Pizer
Assignment 2: **Segmentation by deformable models** — Due in sections

In this assignment you are to compare two deformable model methods for segmenting dark regions in 2D images, namely geodesic snakes using SNAP and active shape models (ASM). The assignment is due in sections.

I. The first section is due on Wednesday, 5 October. In it you are to

1) read the Yushkevich et al. SNAP paper published in *Neuroimage*, as provided on the course blackboard;
2) find the SNAP program in ITK (www.itk.org);
3) create a few simple n×n binary images of a dark ellipse on a light background (for such an ellipse, set all pixels (x,y) such that $[(x-c_x)/r_x]^2 + [(y-c_y)/r_y]^2 < 1$ to 0 and all other pixels to a large value, and put these images in the form usable by SNAP, as follows:

   Take the 2D image and stack it upon itself n times to produce an n×n×n 3D array. Then use the program writeMETA('filename.mhd') (provided in the file META image reader/writer on the blackboard) to produce a meta-format image desired by SNAP.

Run SNAP on these images, showing how it can segment this object. Your method should deform an interactively specified circular balloon onto the ellipse. More precisely, since SNAP is designed to work on 3D images, initialize with a sphere whose cross-section in the middle slice is the circle. Let that initial balloon be partly within the warped ellipse and partially outside of it.

On the result of SNAP apply the program loadMETA('filename.mhd') (provided in the file META image reader/writer on the blackboard) to produce a 3D array, and then pick the middle slice to produce your final result.

Pass in an image of a single result. This section will prepare you to run SNAP on some medical images that will be provided later.

II. The second section involves preparing to use ASM by forming a shape space on a PDM using PCA. It is due on Monday, 10 October. You should write a program that takes in n tuples of N/2 (x_i,y_i) 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. This should applied to the data in a .rar file named "data and programs" on the blackboard, which you can unzip to obtain a data folder consisting of 17 files of points. Each file contains a set of points describing a face. The other two files in the .rar file are two matlab programs for you to use in accomplishing the face translations and displaying the faces. One is the program 'readpoints', which reads the list of points in the data (each file is a face), and the other is a program 'drawFaceParts' which displays results.

drawFaceParts takes two inputs: (pts, 'color and line style'). The first is obtained by a matlab line of the form; pts = readpoints('filename') The second is 'k-' if you want a black line, 'g-' if you want a green line, and 'r-' if you want a red line. If you put multiple drawFaceParts invocations successively, with different point files and styles, you will see the various drawings superimposed. The following is a sample program to draw 3 faces.

```
fName = './dat/107_0764.pts';
```

```
fName2 = './dat/107_0766.pts';
fName3 = './dat/107_0799.pts';
pts1 = readPoints(fName );
pts2 = readPoints( fName2 );
pts3 = readPoints( fName3 );
drawFaceParts( pts1, 'k-' );
drawFaceParts( pts2, 'g-' );
drawFaceParts( pts3, 'r-' );
```

Pass in the mean face with the translated version of one of the sample faces superimposed on it.

Finally, the mean should be subtracted from each aligned tuple, and PCA [you should use the matlab program princomp for PCA] should be applied to the mean-subtracted set. The method should record the first 3 eigenmodes and the associated principal variances (eigenvalues).

III. Write a program to carry out ASM to find a dark region using a given set of eigenmodes and principal variances and an initial cyclic PDM of N/2 2D points. The program will be due on Monday, 17 October. The program will need to compute a normal at each PDM point as the average of the normals to the two piecewise linear segments defined through that point and its respective preceding and following points. Next Monday you will be given medical image data and training PDM segmentations.

This program should begin by computing the 3D gradient vector (not the gradient magnitude) image. To form the 3-dimensional shape space to use in ASM, use the mean and first three eigenmodes of the PCA result computed by the method in part II. That is, along the ith eigenvector $\mathbf{v}^i$ (of length σ_i, the square root of the associated principal variance) the shape space will be $\alpha\mathbf{v}^i$, with α ranging from $\alpha = -3$ to $\alpha = -3$. To compute the image match for any candidate PDM, at each vertex you need to compute the (unit) inward-pointing boundary normal vector and along each normal compute 13 points 1 interpixel distance apart centered at the boundary point in question. Using bilinear interpolation, interpolate the gradient at each of these points, and take the dot product between normal and the gradient to produce the directional derivative in the normal direction. Use this as your geometry-to-image match value. That is, move the point along the normal to that point from the 13 that has the largest value of the directional derivative in the normal direction. After all the points in the PDM are moved, project the resulting PDM onto the shape space to produce the next candidate PDM. Stop the iteration when no point moves more than 1 unit (interpixel distance) of distance or when you have accomplished some maximum number of iterations that you choose.

IV. Compare the two methods, SNAP and ASM, on the test medical images provided. The comparison is due Wednesday, 19 October. In the ASM initialize the ASM with the mean (from the training) with its center at the same place you initialized the circular balloon at in SNAP.

Your result for method 1 will be a binary image. For method 2 the result will be a PDM, which can be plugged into the mean .byu image (see above) to produce a corresponding .byu. You will want to display this .byu against the correct answer binary image (see above) to get a visual evaluation of the segmentation.

You need to evaluate the segmentations quantitatively using the “Dice” measure of volume overlap between the correct answer binary image C and a segmentation binary image S. Dice(C,S) is defined as the volume (# of voxels) in the intersection between C and S divided by the average of the volume of C and the volume of S. You should compute this measure for segmentations you are comparing.

You will compare these two segmentation methods in their behaviors and in their effectiveness. In what you pass in, please give a few examples of target images, and describe the behaviors of each of the methods as the degree of warp changes. Also, pass in the code for your ASM method program.