Computerised Image Analysis MN2, HT02
Exercise 2: Active Shape Models and Snakes

Active Shape Models (ASMs) and Snakes are two types of computer based methods used to find object boundaries. Both ASMs and Snakes use a priori information about the searched shape, and are hence also referred to as knowledge based methods. However, although the end result may be similar, the way ASMs and Snakes achieve the goal of locating an object's boundary is quite different. The purpose of this exercise is to give a hint of the difference between the methods and to give some practical experiences with ASMs and Snakes.

Formality

- I would prefer you to work in groups of two persons. That way, you can discuss with each other and solve difficulties more easily. It should be possible to work from any place where Matlab is available.
- You should prepare a written report, one for each group. Remember when you write the report that it should show that you have learned about the methods you have worked with. I am not that interested in how well you solved the particular problem, as long as you describe precisely what you have understood.
- Hand in the report to Ingela or to me, or send it by email to xavier@cb.uu.se. You are very welcome to write the report in English, but I can read Swedish too, or French if you like ☺.
- As the first part is more passive, I would like you not to spend more than the third of your time on this one, and concentrate more on the second part, that requires more thinking.
- Deadline: at the latest at the day for the written exam.

Part 1. Active Shape Models (ASM)

The ASM package you will use was developed by Ghassan Hamarneh at the Image Analysis Group, Department of Signals and Systems at Chalmers University of Technology, to which I made some personal modifications. The ASM model implements a statistically constrained Point Distribution Model. The user trains the model on several examples by choosing landmark points on a set of example images. A Principal Component Analysis is then derived from the training stage and gives the modes of variation of the model. These parameters are used to help the model fit the final dataset, by allowing model variability only along the principal variation modes. It is based on an article by Cootes et al ("Active Shape Models – Their Training and Application", Computer Vision and Image Understanding, Vol.61, No.1, pp.38-59, 1995) that you can copy from me if you need. The description given in the course book (*Sonka*, pp. 380-390) should nevertheless be enough to do the lab.

Getting started

To start the ASM program:

1. Copy the file http://www.cb.uu.se/~ingela/Teaching/ImageAnalysis/Snakes/asm.tar to your working directory.
2. Unpack the archive (type tar xvf asm.tar). A main directory named asm will be created.
3. Start Matlab
4. Go in the ASM directory and type asm

The ASM program is divided into three main stages:

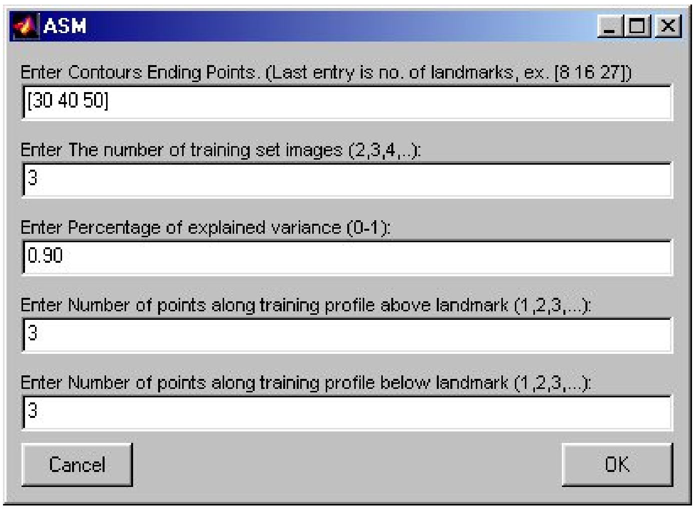
1. Training
2. Trying weights
3. Searching for a shape in an image

Your task is now to go through these steps and apply the ASM model to some image data.

Stage 1: Training

To enter parameters

When you select training, the following dialogue below will be displayed.



Enter Contours Ending Points. (Last entry is no. of landmarks, ex. [8 16 27])
[30 40 50]

Enter The number of training set images (2,3,4,..):
3

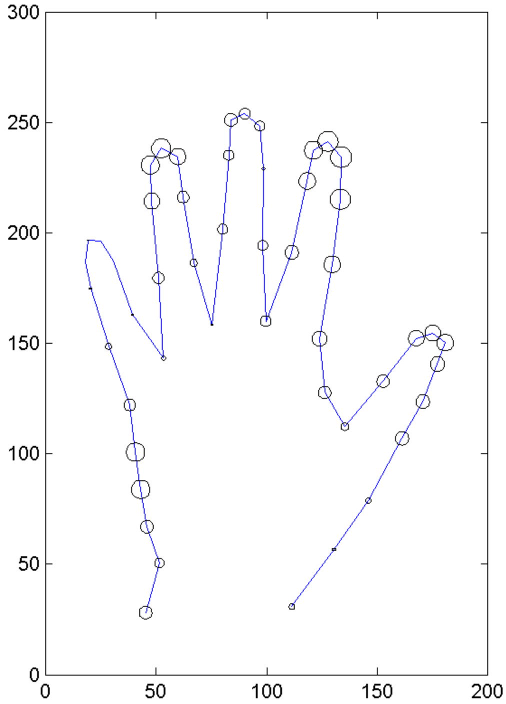
Enter Percentage of explained variance (0-1):
0.90

Enter Number of points along training profile above landmark (1,2,3,...):
3

Enter Number of points along training profile below landmark (1,2,3,...):
3

Some comments:

- You specify how many contours you want, and how many landmarks each contour should have in the field ”Enter Contour Ending Points”. You should enter the last point for each contour. That is, [30 40 50] means that you have three contours, the first having 30 landmarks and while the two others have 10 landmarks each.
- You specify the number of training images in the second field.
- In the two lower fields, you specify how many points above and below each landmark that should be used for training. Note that the optimal choice depends on how the image looks like around the object shape.



Selecting landmarks

After you have selected your parameters and clicked OK, a file selection box is displayed and you should select an image (a bmp-file). The image will be displayed and you then use the mouse to enter landmark.

- Click the left mouse button to place a landmark.
- Move the mouse near a landmark and press the space bar to remove a landmark.

If you choose to work with more than one contour, don't pay attention to the apparent link that appears between 2 different contours that you may draw, only the points are taken into account by the program.

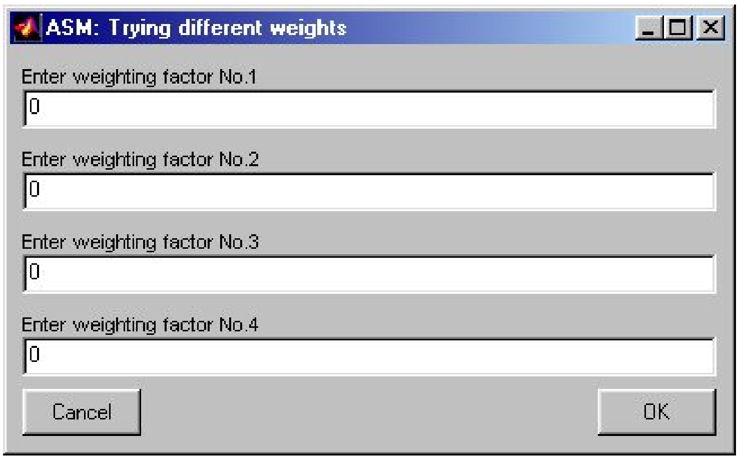
After you have entered the number of landmarks you specified, you will be prompted to select the next training image.

Saving the results

After you have trained on the last image, the training data will be aligned and the ASM model will be computed. You then have the option to save your result. If you choose to save your data, the ASM model will be saved and you can in subsequent session load the data and you don't have to retrain the model. I of course recommend you to do this…

Stage 2: Trying weights

After training, or after you have loaded a saved file, you can look at the modes of variation in your model. To do this, select Try weights and the dialogue below will be displayed. You can then enter different weighting factors and the deformed shape will be displayed.



Enter weighting factor No.1
0

Enter weighting factor No.2
0

Enter weighting factor No.3
0

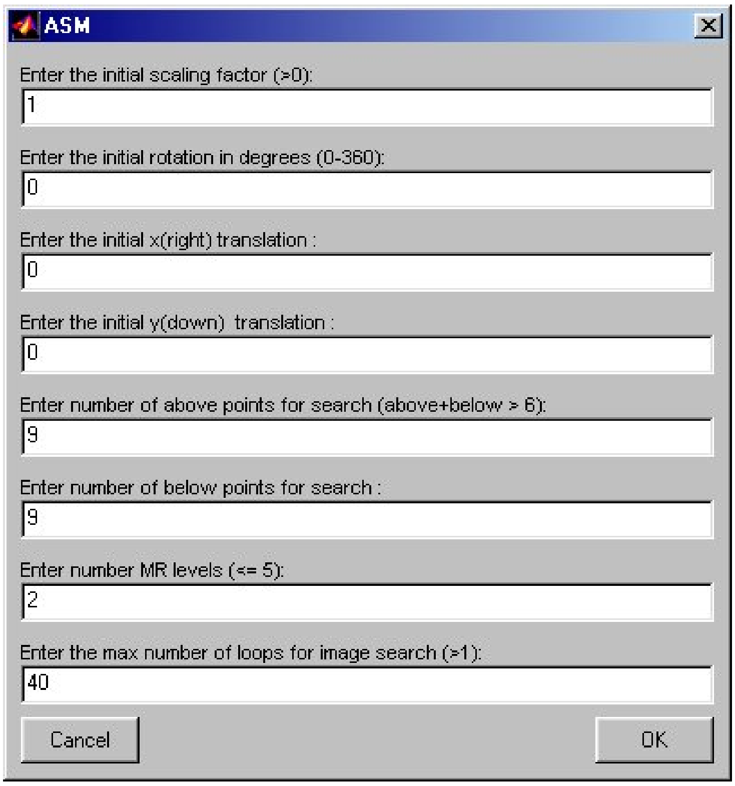
Enter weighting factor No.4
0

Note:

- The weighting factors are given in arbitrary numbers. You can for example try to enter +70 and then -70.
- In this example, four weighting factors can be entered. When you train the model on some new data, you may get more or less weighting factors.

Stage 3: Searching for a shape in an image

In this stage, you can use the ASM model to search for a shape in an image. You will be prompted to select an image and after that, the dialogue below will be displayed.



Enter the initial scaling factor (>0):
1

Enter the initial rotation in degrees (0-360):
0

Enter the initial x(right) translation :
0

Enter the initial y(down) translation :
0

Enter number of above points for search (above+below > 6):
9

Enter number of below points for search :
9

Enter number MR levels (<= 5):
2

Enter the max number of loops for image search (>1):
40

Select parameters and click OK. The image will then be displayed and the contour of the searched shape will be outlined. If the initial position is not close enough, answer no when prompted and adjust the parameters in the dialogue.

Note: *MR levels* means multi resolution levels. If you select 5, you will start the optimisation at a very low resolution. This may cause the contour to drift away from its position close to the boundary. For most applications 2 MR levels work fine.

Your assignment

The dataset you will work with are 7 binary images of right hands from doctoral students at the CBA. All images (hand1.bmp,…, hand7.bmp) are located in the directory '**ASM/images.**

Your task:

1. Train the ASM. To get familiar with the program, you may wish to start with a small structure such as one finger so you just have to enter a few landmarks. Once you are familiar with the interface etc. train the ASM using more landmarks. Try to segment the whole hand.
   Note: It is very important that you should place the landmarks in about the same position in all images. A sketch on a piece of paper can help you to plan the positioning.
2. Save the result to a file.
3. Try weights. Explore the variations of the training set.
4. Try the third stage and try to search for the trained shape in a new image. Since the model was trained on only a few examples, you can select one of the images in the training set. Alternatively, you can train the ASM on 6 images and then apply the model to the 7th one.

As an additional, non-mandatory assignment, you have also, if you have been bored with the binary case, three sets of greyscale images available for testing:

1. head1.bmp,… head6.bmp. These are MRI-images representing axial slices at the level of the cerebellum.
2. corp1.bmp,… corp6.bmp. A close-up of a structure of the brain, the corpus calosum in the sagittal plane

Report on part 1.

Describe what is done during the three stages. That is, relate what is done to the theory about ASMs and describe shortly what is computed etc. during the stages. Give also a short description of what you did, problems you saw, and include an image of the result. Answer the following questions:

1. Training: parameter about explained variance. How is it used in the model?
2. Trying weights: As written above, the weights are not scaled. How would you suggest to scale them in order to allow for the user to change the weights in a proper way (See *Sonka* for help)?

Part 2. Snakes

Active contours - or snakes - is a shape representation technique that can be understood as computer-generated curves that move within images to find object boundaries. Deformation of the snake towards the desired position is driven by internal forces (elasticity, elongation,…) and external forces, computed from the image data. In this exercise, you will experiment with a snake tool developed at the Image Analysis and Communication Lab at The Johns Hopkins University (for more details, see http://iacl.ece.jhu.edu/projects/gvf/). In addition to basic snake manipulation tools, it implements an effective external force field: the Gradient vector Flow. I recommend, to get a good knowledge on snakes in general, and gradient vector flow in particular, to read the first part (pp 131-145) of the chapter 3 of the excellent book "Handbook of Medical Imaging", Sonka and Fitzpatrick ed. This chapter has been written by the authors of the method, and is available for download at the above Internet address. You may also want to read the course book chapter on snakes (pp. 374-380)
In this part, you have a little more freedom than in Part I. You can, to achieve your task, modify the existing modules, and/or add new functionalities that you judge useful.

Getting started

1. Copy the file http://www.cb.uu.se/~ingela/Teaching/ImageAnalysis/Snakes/snake.tar to your working directory.
2. Unpack the archive (type tar xvf snake.tar). A main directory named GVF and several subdirectories (snake, examples…) will be created.
3. Start Matlab.
4. Add a search path for Matlab to the snake directory. Type addpath('**your_path/GVF/snake'**).
5. In Matlab, Go to the snake directory.

The snake program

A number of modules are available and using these, you can create your own applications. Type ”help snake” to get the description of the modules included:

Image input/output

Rawread	- Read a Portable Bitmap file, or a raw file
Rawwrite	- Write a Portable Bitmap file, or a raw file

Image Display

imdisp	- Display an image

Active Contour

Snakeinit	- Initialise the snake manually
Snakedeform	- Deform snake in the given external force field
snakedeform2	- Deform snake in the given external force field with pressure force
snakedisp	- Display a snake
snakeinterp	- Interpolate the snake adaptively
snakeinterp1	- Interpolate the snake at a fixed resolution (better implemented than snakeinterp)

Gradient Vector Flow

GVF	- Compute the gradient vector flow field

Other

dt	- Simple distance transform
gaussianBlur	- Blurring an image using Gaussian kernel
gaussianMask	- Generate a discrete Gaussian mask

Your assignment

1. Run the demos and answer briefly to the corresponding questions.

- tradition_ex.m : what is the "traditional force" in this case?
- distance_ex.m : what is the advantage of the method in terms of "attraction range"?
- balloon_ex.m : what is the main problem with the pressure force (balloon) approach?
- gvf_ex.m : why does the snake manage to reach the bottom of the inner part of the U-shape image?

You can see by running these examples what the algorithm is able to achieve, and what tools you have at hand.

2. Run the program mri.m.

The idea is that the snake should segment the brain surface, but it will not succeed. Modify the code (change and add functions), explore different settings of parameters and see if you can make the snake perform better.

The following hints might help you to begin with. Think of the 3 different parts of the segmenting process using GVF snakes. You may work to improve each of these:

- Initialisation of the snake.
- Parameters relative to the internal force of the snake.
- Type of external force used, and data used to create it.

Report on part 2.

Write a short summary describing what you did. How did you solve the problem of finding the border of the brain? What functions did you try? Please include a picture of the result segmentation.

Finally, your comments on the exercise will be much appreciated.