Figure 1: *The sdemo splash screen and first menu*

3 GVF Snakes

In this session you will study the strengths and weaknesses of Gradient Vector Flow snakes. Read the paper “Snakes, Shapes, and Gradient Vector Flow” by Xu and Price as background material.

3.1 Starting the session

For this practical you must first download file “snakeprakt.zip” from Nestor to your work directory, and type

```
unzip snakeprakt.zip
```

Then type:

```
cd SnakePrakt
```

and start MatLab. You can now start a demo of GVF snakes and classical snakes by typing

```
sdemo
```

The version you will be using is actually an updated version, but the splash screen remains the same. After clicking “continue” The actual demo starts as shown in Figure 1 The “open” button allows selecting different images. The “examples” menu lets you choose between the two examples, you can select smoothing strength by changing the value of sigma (Gaussian smoothing, not advised by the authors), and the “next” button moves you on to the next stage. When using the example images U64.pgm and “room.pgm” you should not have the gradient box checked!

In the second menu of sdemo, you can select which kind of snake you want to choose: classical or GVF snakes. It also allows you to go back to image and smoothing parameter selection. When choosing GVF snakes two parameters can be altered (help on these parameters is available in the scrollable text box, if rather small).

From here we move on to the actual computation of the vector field (which can take some time), and the third menu. In the latter menu you can select various parameters concerning the snake itself. The “deform” button starts the iteration process. In this case the snake iterates until the desired number of iterations has been reached, not when it has stopped moving. Clicking “Deform” again does not reset to the initial position, but simply continues the iteration. You can reset to the initial position, using the “Initial” button. You can select either a circle or a free-hand manual initialization. In the latter case you must click the “Manual” button, and click as many point as needed in the original image, and then click “End M.” as the “Manual” button will label itself. Snake positions can be loaded using the “Load S.” button, and these may also serve as initial position

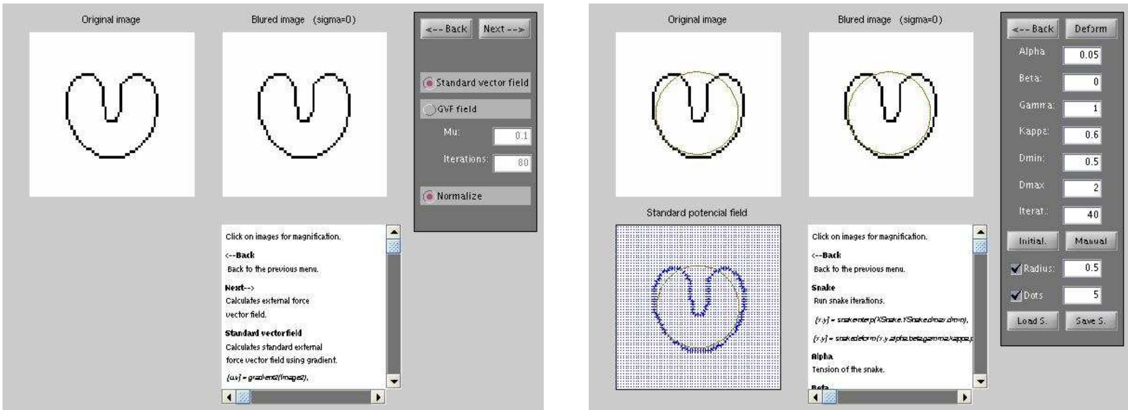

Figure 2: *The next two screens of sdemo*

You can export figures containing the snake by clicking on the images. This pops up an enlarged image in a new figure, which can be saved through the file menu in the usual way.

3.2 EXCERCISES

Exercise 1.

Starting with the U-shape, try both GVF and classical snakes in various parameter settings. Note that you may need far more iterations than the 40 which is used as a default value. Try to find the most important differences in behaviour. Is GVF better? Motivate your answer. Given enough iterations, do classical snakes converge into the “dip” in the figure?

Exercise 2.

Try the same with the “room” example. Can you get GVF to fit this time? How does this compare to classical snakes?

Exercise 3.

Load `chest.pgm` and try to segment the left and right lung as well as possible. You can find fairly good initial positions in `mychest.mat` and `mychest2.mat` (in the `images` subdirectory. Can you segment these features well using either type?

Exercise 4.

Try the same with `new.pgm`. Contrast is higher, does this help?

Exercise 5.

Now use `heart.pgm`, a heart ultrasound image with a far worse signal-to-noise ratio. Experiment with different values of sigma and different initializations (e.g. heart.mat, heart1.mat).

Exercise 6.

Based on the above experiments, what is your opinion of snake-based segmentation. What are the key problems when using these techniques. Could the two methods tried here be combined in some way to get better results? If so, demonstrate this using e.g. the “room.pgm” example.