Computer Assisted Image Analysis II
Spring 2016

Excercise 2: Snakes and Image Registration

Snakes is a computer based method used to find object boundaries. A snake can be described as a flexible line or closed contour that evolves over time to some minimal energy configuration, hopefully converging on boundaries in the image.

Image registration is the process of transforming one image (commonly known as the 'floating image') to align it to another image ('base image'). The transformations can include translation, rotation, scaling, warping, mirroring and so on. Because of the many possible degrees of freedom image registration is often a computationally heavy operation.

The purpose of this exercise is to give you the opportunity to try using snakes as well as solving an image registration problem.

Every part of the exercise has some instructions, and sometimes also some questions. Read these **before** *actually performing the part. Answer the questions explaining* **why** *you got a particular result, rather than* **how**. *If you get stuck, do not spend too much time on a particular problem, it is better to continue and ask us later on.*

Formality

- You need an account for the computer systems at the Dept. of Information Technology.
- You should work together in groups of two people. It may be advantageous to have someone to discuss issues with.
- You are requested to prepare a written report in pdf format of the exercise answering all questions including explanatory illustrations. Only one report needs to be turned in per group with names of persons mentioned in it.
- The report is handed in via the Student Portal under 'File areas' → 'Lab reports'. Make sure that you belong to a group before starting the lab. You join a group in the Student Portal by going to 'Group divisions' → 'Lab/Project groups'.
- Status of your reports will be found at the Student portal under 'Progress' → 'Progress, mandatory tasks'.
- **Deadline: 18, Feb 2016**

Getting started

In a lab at the Dept. of Information Technology running Windows: Log on to one of the workstations. MatlabR2013b is found under `ITC-Application Explorer(Zenworks)`

Download the lab material from the Student portal, 'File areas' → 'Lab material'

1 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 by Cris Luengo for Matlab. In addition to basic snake manipulation tools, he has implemented two effective external force fields: the Gradient Vector Flow and the Vector Field Convolution. To read more about GVF go to the following link `http://iacl.ece.jhu.edu/projects/gvf/`. To read more about VFC go to `http://viva.ee.virginia.edu/research_vfc.html`. Also, make sure you understand the course book chapter on snakes (Sonka [2], Section 7.2).

For this assignment you can, to achieve your task, modify the existing modules, and/or add new functionalities that you judge useful.

Set up

Change the directory to `your_path/Snakes/`.

Assignment instructions

1. Start off by running the examples (`example1.m, example2.m, example3.m`). Make sure you understand what happens in each example and then answer the following questions:

   - How are the external forces in the images calculated?
   - How do the external forces differ in terms of "attraction range"?
   - What is the main problem with the pressure force (balloon) approach?

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

2. Run the file `exercise1.m`. This will give you a snake segmentation of a synthetic image. Use this example to play with the initialisation of the snake, the snakes internal forces and the snakes external forces.

3. Run the file `exercise2.m`. You will see a failed attempt at segmenting one of the pears in the image. Make any changes to the code you see as necessary to achieve a better segmentation. The result should look something like figure 1.

   The following hints might help you to begin. You may need to improve each of these:

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

Report instructions

Write a short summary describing what you did. How did you solve the problem of finding the border of the pear? What functions did you try? What was the result of different parameter adjustments? Please include a picture of the resulting segmentation.

Figure 1: Target result for snake exercise

2 Image Registration

Exhaustive search

As a first assignment you will study a pre-written example of an exhaustive search approach for image registration. Change the directory to `your_path/Registration` and open the file `ex_registration_exhaustive.m`. Run the code and while it runs go through the code line by line and make sure you understand what is happening. Do not forget to look closer at the functions that are called (e.g. `exhaustive_match.m`). The result is illustrated in Fig. 2.

Now answer the following questions:

1. Which similarity measure is used in the example?
2. What information in the two images is compared?
3. Translation and rotation in the example are always performed in a certain order. What is that order and why do you think it is done this way?
4. How do you locate the optimal transformation in the score space for this example?

Chamfer matching

Now we will take a look at a gradient descent approach to image registration. The biggest issue with the exhaustive approach is that it (as you now know) takes a lot of time to compute. The time consumption increases exponentially with each added degree of freedom.

A different approach to the registration problem is to use a method know as **Chamfer matching**. For this method the similarity measure is the distance between two binary images (generated using e.g. thresholding or an edge detector such as Canny). The perhaps easiest way to implement this is illustrated in Fig. 3. The binary image (in this example the edge map) of the floating image is placed on top of the distance map generated from the binary base image. The sum of the distance values from the overlap is used as the score for that specific transformation.

Figure 2: **Top:** Three map sections from three different map services. All sections have overlapping regions but they have different rotations. Leftmost image is set as base image. **Bottom:** Result of a registration process.

Instead of evaluating all possible transformations in the allowed transformation space, Chamfer matching can be used with a gradient descent approach to find an optimal value. Given a certain initial transformation (e.g. a certain position and rotation), a number of predetermined transformations (e.g. move a little to the left, rotate clockwise 5° etc.) are evaluated. The transformation that yields the lowest distance is then chosen as the starting point for the next iteration. This behaviour is repeated until a predetermined stopping criterion is met.

Your task is to implement a Chamfer matching algorithm that is able to match the three map segments. As help you are given the file `chamferDemo.m` which basically solves the problem described in figure 3.

For your task you should use `01.png` as the base image. The function `meanRGB` can be used to merge up to three images so you can see how well you have succeeded. When you are done you should have something similar to the result in Fig.2.

For the report you should answer the following 5 questions:

1. Which image transformations are you evaluating for the floating image at each iteration and why?
2. What stop criterion have you chosen and why?
3. Supply a plot showing how the distance measure decreases for each iteration (see Fig. 4).
4. How do you locate the optimal transformation in your score space?

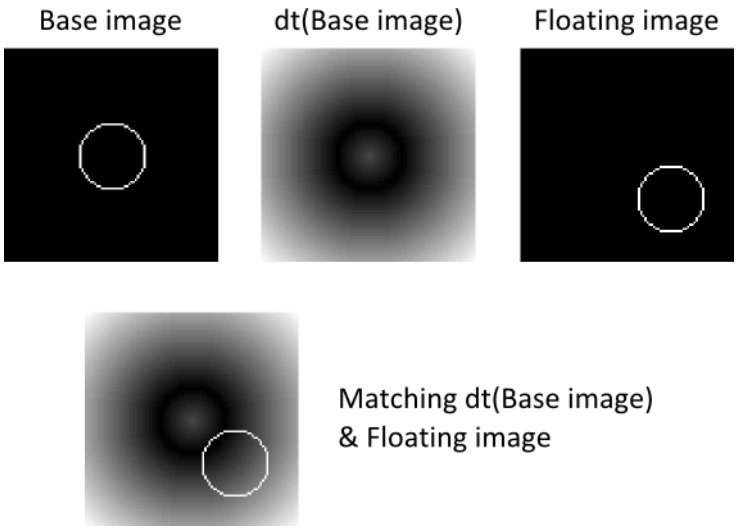

Figure 3: Crucial steps of the Chamfer matching process.

5. How would you avoid getting stuck in local minima (see Fig 4)?

Extra task!

If you feel that you cannot get enough of Chamfer matching here's a challenge for you. Try matching the image 04.png to any of the other images. Slightly more difficult but definitely possible.

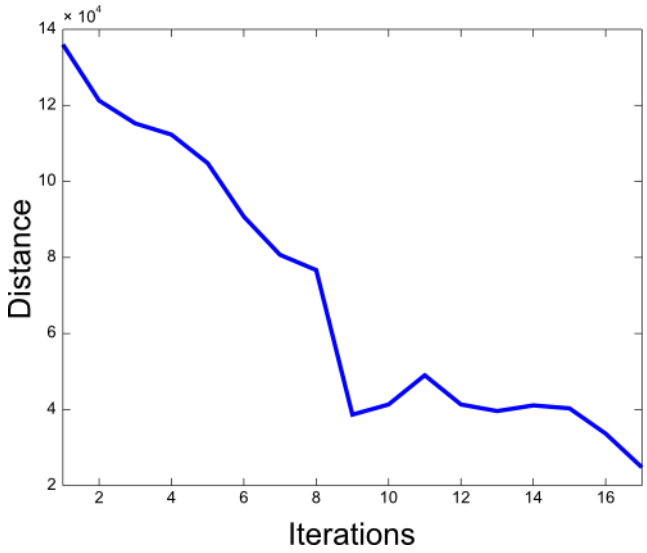

Figure 4: Plot showing the Chamfer distance against iterations in the registration process.

References

[1] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, Inc, Upper Saddle River, New Jersey, 2nd Ed, 2002.

[2] Milan Sonka and Vaclav Havlac and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson Learning, 3rd ed, 2008.