Computer Assisted Image Analysis MN2 HT 2008

Fuzzy Connectedness Segmentation

Formality

- The task will be carried out in Matlab. It is preferable that you work together in groups of two. It may be advantageous to have someone to discuss issues with.
- You are requested to prepare a written report (one per group) of the exercise with explanatory illustrations. No hand writing or hand drawing, please, unless very clear. **I want *commented* result images in the report!**

  Possible ways to hand in the report:

  – Send as an attachment of an e-mail message to `joakim@cb.uu.se` in any format I can read (pdf, PostScript, . . .).
  – Prepare a web-page and simply send the web address in an e-mail.
  – Print the report and leave to me (Joakim) in person or in my mailbox at CBA.
- **Deadline:** As always, the sooner the better, but the latest at the day for the written exam.
- Status of your reports will be found at the web-page:
  `http://www.it.uu.se/edu/course/homepage/bild2/ht08/`

1 Absolute Fuzzy Connectedness

The articles referred to in this assignment can be found at `http://www.it.uu.se/edu/course/homepage/bild2/ht08/Exercises/fuzzy/`.

Download and unzip `http://www.it.uu.se/edu/course/homepage/bild2/ht08/Exercises/fuzzy/FuzzConn.zip` Content is a number of matlab files.

- The file **afc.m** provides an implementation of absolute fuzzy connectedness according to kFOEMS in [2]. Do read and compare with the code presented in the lecture notes `http://www.it.uu.se/edu/course/homepage/bild2/ht08/Lectures/lecture11.pdf`.
- The files **adjacency.m** and **affinity.m** computes adjacency and affinity according to [3]. This is similar to what is presented in the lecture note.
- The file **fctest.m** gives an example of how to use the provided functions.

Start matlab in the FuzzConn directory and run **fctest**. Study the code of **fctest.m** and the output. Notice that any path starting from the seeds is monotonically decreasing away from it.

Try changing/adding seed points (S) by editing **fctest.m** and see what happens.

Q1 Compute a fuzzy connected component C_1 of strength Θ from some seed S_1. Pick other seed points S_2 all placed within C_1 and compute a new connected component C_2 of strength Θ from the new seeds. How does C_2 look in comparison with C_1? Say something about robustness with respect to seed point selection.

Q2 Increase the size of what is locally adjacent by increasing n to 3 and see what happens. Try to motivate when it can be useful have a larger adjacency than the nearest 4 or 8 neighbours.

Use the same method to segment a grey-scale image of your choice. Do not pick a too large image (i.e., more than around 256x256), since the matlab implementation provided is more aimed at pedagogic use than speed.

Q3 How did it go? Include images and comments in the report.

2 Iterative Relative Fuzzy Connectedness

The file **irfc.m** provides an implementation of iterated relative fuzzy connectedness according to kIRMOFS in [1]. That function calls **afc.m** a number of times to partition an image into components defined by multiple seeds from at least two classes. Classes are given by seed numbers starting from 1.

Modify **fctest.m** (copy it to a new name first) to compute IRFC instead of AFC. The iterations makes it even slower, so do not use too large images.

Again, try to segment an image of your choice.

Q4 How did it go? Include images and comments in the report.

3 Modification

Perform at least one of the following tasks. You are welcome to do all three of them if you like.

1. Usually there is limited interest in computing connectedness values to all pixels. If performing segmentation by thresholding the FC-map at a level K, then there is no reason to compute lower connectedness values. Since **afc.m** computes the FC by propagating outwards from higher towards lower FC values, we can stop it early.

   Change **afc.m** so that it takes one additional parameter defining when to stop further FC computations.

   Q How did you change the code?

   Q How big speed improvement did it lead to on, e.g., an unmodified **fctest.m**?

2. *Warning: This task probably requires some experience of matlab indexing.*
   The provided code only works for grey-level images. Modify **affinity.m** so that it works for colour images.

   Q How did you change the code?

   Q Give and example on segmentation of a colour image.

3. Compute an FC-map on an image with a smooth gradient from light to dark with a seed at the bright end.

   Q What is the result?

   The provided affinity only considers intensity differences, i.e., image gradients. Often one wish avoid connectedness to lead too far away from the main intensity of the seed region. This can, e.g., be achieved by scaling down affinity by the difference between the mean of the two pixel intensities, and the mean of the seed region intensities. Implement this in **affinity.m** and try again on the smooth ramp image as well as some additional real image.

   Q How did you change the code?

   Q How did it work?

   This approach of estimating some local properties from the seed region, and then letting that affect the affinity, is a very powerful way to adjust the FC to the specific segmentation task.

Good luck! and I hope you will enjoy the task.
Joakim

References

[1] K.C. Ciesielski, J.K. Udupa, P.K. Saha, and Y. Zhuge. Iterative relative fuzzy connectedness for multiple objects with multiple seeds. *Computer Vision and Image Understanding*, 107:160–182, 2007.

[2] P.K. Saha and J.K. Udupa. Fuzzy connected object delineation: axiomatic path strength definition and the case of multiple seeds. *Computer Vision and Image Understanding*, 83:275–295, 2001.

[3] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and application in image segmentation. *Graphical Models and Image Processing*, 58(3):246–261, 1996.