TUTORIAL 5: SEGMENTATION

Lecturer: Juan Nieto
Tutor: Christopher Brunner

This tutorial is MARKED. Due date: Tuesday after the lab.

Instructions

- This tutorial will be evaluated. This is individual work: each student has to complete his/her own report.
- A printed copy of this report is to be handed in and a .pdf copy and code is to be emailed by the start of the lecture, Tuesday 10th September, 2013.
- A title page must include: AMME4710 Image Processing Tutorial, student name and SID, email address. The electronic version MUST be as a .PDF and be titled LASTNAME_SID_AMME4710_Tutorial4.pdf.
- The email subject MUST BE: **AMME4710 Tutorial 5 – Segmentation**.
- Answer questions briefly and clearly. Ensure that you include the correct title for each question. I strongly suggest you answer each question under the headings Aim/Introduction, Method, Results, Discussion for clarity.
- "Aim/Introduction" should quickly say what you are doing and why it is an important part of computer vision.
- "Method" needs to be explained briefly using flow charts and/or the theory involved (i.e. not using code or saying "followed instructions" – show you understand what you are doing and how the theory/code works in general.)
- "Results" shows your images and other results with captions explaining what it is (relate it to the method). You can include brief descriptions of what the results are showing.
- "Discussion" discusses, explains and expands on the results.
- If you are displaying images, please make sure the image appears as you expect it to in the hard copy and the pdf. Please use appropriately sized images, and remember that if you are comparing images, make them the same size and put them next to each other.

Marking Scheme:

Format (clarity, grammar, image size, captions, white space etc): 10
Q1: 10
Q2: 10
Q3: 10
Q4: 30

Note this week that the bulk of marks are for your evaluation of the methods in part 4. Therefore, be brief in Q1, 2 and 3. As usual, do an introduction and method, but the results for these 3 questions can be shown for a single (appropriate) image.

Introduction

This tutorial works through some of the approaches for image segmentation discussed in the lecture.

There are four main parts to this tutorial.

- Active contours (snakes)
- Mean shift
- K-means clustering
- Graph cuts

We use functions from the Matlab Image Processing Toolbox and Svoboda's book [Svoboda et al., 2008]. Use the Matlab help to get more details on the functions and their parameters. Review the lecture slides for more information on the principles behind the algorithms.

You should experiment with this tutorial yourself, by changing the parameters while testing different images. Choose from the pictures available or use your own pictures. Starting with default values, the parameters need to be tuned with the objective of improving the segmentation as much as possible.

1. Snakes

We use an implementation of the snakes algorithm based on the balloon-force snake by [Cohen et al, 1992]. This approach modifies the definition of external forces (derived from the gradient of the image) presented in the traditional snake. A new pressure force is introduced which makes the curve behave like a balloon. In this model, contour moves out (inflation) or in (deflation) along its normal thus eliminating the need for the initial curve to be close to the solution (correct edges) to converge. The Balloon Snake passes over relatively weak edges and it stops on salient edges. By placing balloon snake inside/outside the object of interest and inflate/deflate contour to its boundary, this new model is useful for objects that are difficult to locate or too complex to trace by the traditional snake.

Step through and examine carefully the code in snake_demo.m and the actual algorithm in snake.m. There are two examples of balloon snake in this demo, contour expanding and contour contracting. The second example also shows how to deal with colour images. Note that you can try other approaches for converting images to grayscale.

2. Mean shift

Use the meanshsegm.m code to segment images using the mean shift algorithm. An example is given in meanshsegm_demo.m. In this implementation by [Svoboda et al., 2008] the joint density is estimated with the kernel

$Kx=ckExshs2+xrhr2$,

where kE is the Epanechnikov kernel profile, $xs$and xr are the spatial and range parts of the image vector, and hs and hr are the corresponding kernel sizes. Note that this

Matlab implementation is not optimized, segmenting a small-size colour image (200x300 pixels) can take a few minutes. The size of the kernels also affect processing time.

3. K-means clustering and Graph Cuts

We first demonstrate the use of k-means clustering. Then, we use the results as initial estimates for graph-cuts segmentation. Since a Matlab implementation of graph-cuts would be painfully slow, we use a C++ implementation [Kolmogorov et al., 2004] with a Matlab wrapper by S. Bagon. Before running the demo you need to compile the code using compile_gc.m.

Step through and examine carefully the code in kmeansgraphcuts_demo.m. Test different parameter options for the kmeans.m function. Note that the number of clusters must be chosen in advance for each image.

We use the cluster centres found by k-means clustering to define the affinity matrix for the graph cut algorithm. The Euclidean distance is used as metric. Implement the Mahalanobis distance instead and test if it improves the results. The Mahalanobis distance is defined as $x,y=x-ytS-1(x-y)$, where S is the covariance matrix.

In your discussion:

Did the Mahalanobis distance provide any improvement on the graph-cut results?

Why do you think adding edge information improves the results of graph-cuts?

4. Overall Evaluation and Discussion

Compare the results of each algorithm above on five test images.

Objectively compare the results of each method.

Comment on what types of images are handled better by different methods. (i.e. if I was to give you a random picture and ask you to segment it, what features in the image would push you towards choosing one method over another).

Rank the methods according to segmentation quality, speed and ease of use.

Which method was superior? Why? Are there exceptions?

Briefly describe (using your own words) the why this method is superior to the other methods (this is not a comment on the results, this is a description of how the system works mathematically compared to the other methods.)