Centre for Mathematical Sciences, Mathematics, LTH, September 2014

# Image Analysis, Assignment 4

In this assignment you will study segmentation using graph cuts and continue on your Optical Character Recognition system. The data for the assignments is on the course homepage:
`http://www.ctr.maths.lu.se/course/newimagean/2014/`

The assignment is due on thursday of study week 6. Make sure you answer all questions and provide complete solutions to the exercises. You may hand in hand-written solutions and/or printouts in the slot marked "bildanalys" on the 3rd floor of the mathematics building and/or as a pdf by mail to fman20@maths.lth.se. Write your name and the assignment number in the subject line. For each exercise, we have tried to make it clear what should be included in the report. In addition, all the code should be submitted as m-files by mail. Make sure that your matlab scripts are well commented and can be executed directly (that is, without loading any data, setting parameters etc. Such things should be done in the script).

You will have time to work with the assignments during the computer laboratory sessions and the exercise sessions. These sessions are intended to provide an opportunity for asking questions to the lecturer on things you have problems with or just to work with the assignment. During the laboratory sessions you should work on the exercises marked "Computer Exercise". The rest of the exercises are intended to provide hints and prepare you for the computer exercises. You are expected to have solved these before you go to the lab sessions. The report should be written individually, however you are encouraged to work together (in the lab session you might have to work in pairs). Keep in mind that everyone is responsible for their own report and should be able to explain all the solutions.

## 1 Convergence of the K-means algorithm

From analysis of functions in one variable we learned that (see for instance page 155 in Persson-Böiers: Analys i en variabel or page 181 in Månsson-Nordbeck: Endimensionell analys.) 'every decreasing real valued function that is bounded from below converges'. This fact can be used to show that the value of the k-means function converges when using the k-means algorithm.

**1)** Prove that the k-means algorithm converges in a *finite* number of steps.

**Hint**: In every step of the k-means algorithms, the correspondences (the vector $c$ in the lecture notes) are updated. Explain why there is only finitely many $c$. In an iteration two things may happen. Either the goal function is strictly less and then $c$ is changed or the goal function does not change and then $c$ does not change (or at least one can choose not to change $c$).

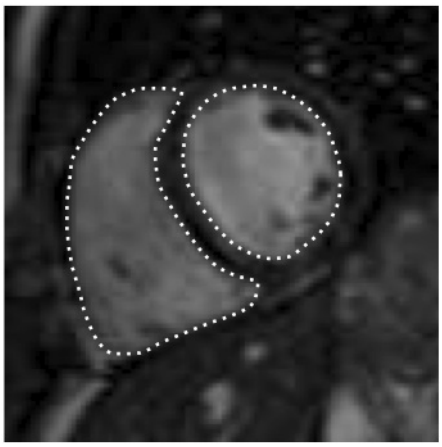

Figure 1: The data given in the task. A slice of a heart imaged by a MRI camera. This view is known as the "short-axis view" taken in a direction where the left and the right heart chambers are visible at the same time. The two chambers are shown inside the dotted lines, the left chamber is to the right in the image and vice versa.

## 2 Segmentation with Graph Cuts

You are given the task of segmenting out two heart chambers in an image by Graph-Cuts. In `heart_data.mat` you are given a number of intensities which have been observed inside the two chambers (*chamber class*) and a number of intensities observed in the background (*background class*). You are also given the image of Figure 1, denoted by $f$ below.

**Hint:** You can use the matlab routines given in laboratory session 3 to solve the min-cut problem. You will need to study the material in the lectures and also read chapter 5.5 in Szelisky "Computer Vision: Algorithms and Applications". You can then prepare by doing the 'Image Segmentation' part described in `lab3en.pdf` on the homepage.

**1)** Assume that the pixel intensities in the two classes are generated by two different Gaussian distributions. In order words, for pixel $i$, the likelihoods of observing intensity $f_i$, $P(f_i \,|\, \text{chamber class})$ and $P(f_i \,|\, \text{background class})$, respectively, are Gaussians. Recall, a Gaussian is given by $P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. Estimate the mean and the standard deviation for these two distributions.

**2)** Construct a graph of the heart image with a data-term consisting of the negative log likelihoods for the two classes. Solve it via max-flow/min-cut. Experiment with the prior/regularization weight (denoted by $\nu$ in the lectures) in order to obtain a reasonable segmentation. (Note: it is hard to get a perfect segmentation.)

**3) Optional.** Fine tune the segmentation by adding ground truth. This can be done by changing the data-term to $\infty$ (or a large number) for the pixels you know which region they belong to. This will force certain pixels to belong to a particular class. Such pixels can be obtained by manual inspection of the image.

**Notes:** Both 2) and 3) can be seen as adjustments a Doctor could perform. In 2) a slider could be added to give interactive response to different weights. Then a pen tool can be used to introduce the ground truth of 3).

In the written solution to these problems, supply both code (e.g. matlab code) and a printout of the results of using your algorithm, i.e. supply examples of input data (e.g. as an image) and result after applying your segmentation algorithm (e.g. also as an image).

# 3 OCR system construction and system testing

Here we will continue the work on our OCR system. You have now a system based on

- `S = im2segment(Im)` - a segmentation algorithm that takes an image $Im$ to a number of segments $S$.
- `x = segment2features(S)` - an algorithms for calculating a feature vector $x$ from a segment $Sk$.
- `y = features2class(x)` - an classifier that takes a feature vector $x$ and calculates a class $1, \ldots, 26$.

The system can then be written as a function

```
function text = myocr(im,classification_data);
% text = myocr(im) - OCR system that takes an image
%    as input and produce a text string as output.

S = im2segment(im); % Segmentation
nrofsegments = length(S);
text = char(zeros(1,nrofsegments));
alfabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
for k = 1:nrofsegments;
        % For each segment ...
    x=segment2features2(S{k});
        % ... calculate feature vector x
    y=features2class(x,classification_data);
        % ... classify feature vector as class y
    text(k)=alfabet(y);
        % ... interpret y as a character
end
```

You have also a script for benchmarking the whole system `benchmark.m`. If all works well you can see how well the system works on real data. However, if for some reason it does not work, it might be difficult to determine which part of the system fails. Is it the segmentation part that fails? Are the features not good enough? Is it the classifier that is not working correctly? To do this it is often useful to both benchmark and to visualize the intermediate results.

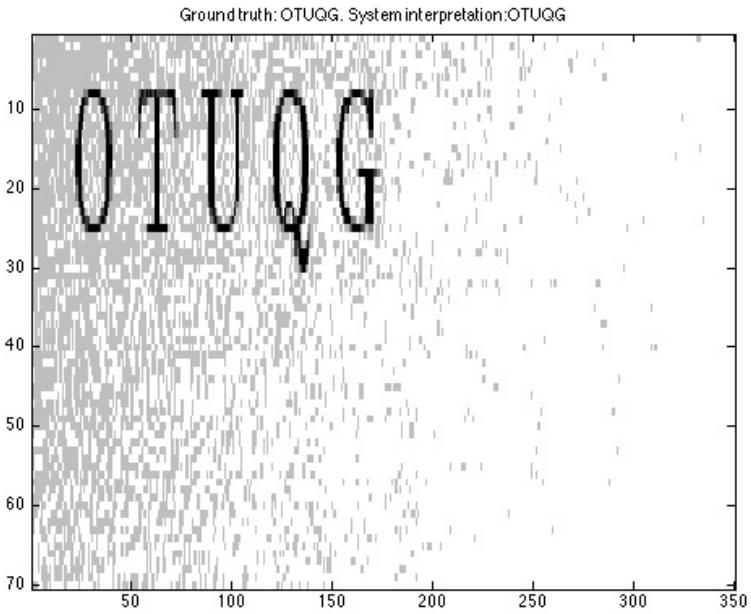


Figure 2: One image with corresponding ground truth and system response.

For this we have constructed another script for benchmarking and visualization `benchmark3.m`. This script needs to be able to run the individual parts of the oct system, i.e. `im2segment`, `segment2features` and `features2class`.

The way we are going to do this is as follows. We store the function names of the different parts of your system in a variable

```
mysystem.segmenter = 'im2segment'; % What is the name of your segmentation-algorith
mysystem.features = 'segment2features2'; % What is the name of your features-algori
mysystem.classifier = 'features2class'; % What is the name of your classification-a
load classification_data
mysystem.classification_data = classification_data;
```

Choose a dataset for testing

```
datadir = '../datasets/home1';
```

Then use `benchmark3`

```
[hitrate,confmat,allres,alljs,alljfg]=benchmark3(mysystem,datadir);
```

to test and visualize to first test your system from Assignment 3 on the three datasets (short1, short2, home1). The script shows the segmentation and interpretation of the system and compares it with a ground truth segmentation and interpretation. The variable `allres` is a matrix where `allres(i,j)` is one if character nr $j$ is correct for image $i$ in the dataset. Similarily the variable `alljs` is a matrix where `alljs(i,j)` is measure on how

well the segment for character nr $j$ is similar to corresponding ground truth segmentation for image $i$ in the dataset. Finally the variable `alljfg` is a matrix where `alljs(i,1)` is an overall score on how well the foreground (the characters) as a whole has been segmented.

- Test your system version 1 from Assignment 3 on the three datasets (short1, short2, home1). Print the overall hitrate for version 1 on the three datasets in the report and describe with your own words the results in your report. If it went well, describe why. If it didn't go well describe why.
- Tro to change your system and produce a version 2 that works better on the three datasets. Print the overall hitrate for version 2 on the three datasets in the report and describe with your own words the results in your report. If it went well, describe why. If it didn't go well describe why.

In the written solution to the segmentation problem, supply both code (e.g. matlab code). Provide error rates for your system (both version 1 and version2) on the three datasets. Discuss the results.