Computer Assisted Image Analysis II
Spring 2016

Excercise 3:

Graph Cuts & Motion Tracking

In this exercise you will be given the opportunity to explore how to binarize an image using the Graph cuts. Furthermore, you will have the opportunity to solve a motion tracking problem.

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: 07, Mar 2016**

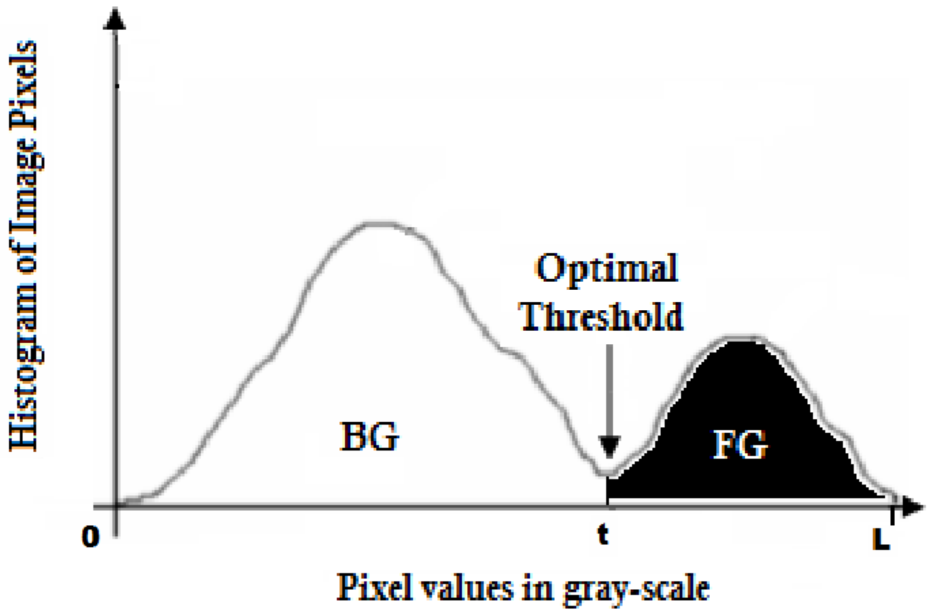

Figure 1: Bi-modal histogram of classes.

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 Binarization using Graph Cut

A binary image is a digital image that has only two possible values for each pixel. Most of the image processing procedures require the image to be available as a binary image, thus making it a common pre-processing stage. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

1.1 Otsu Binarization

We try to understand the fundamental binarization scheme in this exercise. Lets assume that the pixels of an image with N pixels and L intensity levels can be separated into two classes C_0 and C_1 by a threshold t, so that intensity levels *[1, 2, ..., t]* belongs to C_0 and *[t + 1, t + 2, ...,L]* belongs to C_1. The histogram of the image counts the frequency n_i of pixels with intensity levels i = *1, 2, ...,L.*

By normalizing the histogram we can regard it as a probability density function, describing the probability p_i for a pixel to assume intensity level i. Otsu binarization tries to find this optimal threshold by maximizing the inter-class variance, which in turn can be shown to be minimizing the intra-class variance.

1. ***In Ex01.m the input is read from the* DIBCO *folder and the output is written to* Otsu *folder. What would be the ideal binarized image in each of these cases as opposed to what you observe in the output images?***

1.2 Binarization using Graph Cut

The file `Ex02.m` provides an implementation of binarization using Graph Cuts [1]. This exercise needs a mex compilation to run as it interfaces *MATLAB* code with *C* code. Run the following commands in the *MATLAB* command window

I. **mex -setup**. This lets you identify all available compilers that could be used by *MATLAB* to compile *C/C++* code. Interactive questions need to be answered as follows

   (a) *Would you like mex to locate installed compilers [y]/n?* **y**

   (b) *Select a compiler: [1] Microsoft Software Development Kit (SDK) 7.1 in C:/Program Files (x86)/Microsoft Visual Studio 10.0 [0] None. Compiler:* **1** (NOTE: The compile can be gcc depending on the operating system like Linux or the environment like Cygwin. But mostly the preferred choice will be the compiler identified as 1.)

   (c) *Please verify your choices: Compiler: Microsoft Software Development Kit (SDK) 7.1 Location: C:/Program Files (x86)/Microsoft Visual Studio 10.0 Are these correct [y]/n?* **y**

II. **mex imgcut.cpp maxflow.cpp graph.cpp**

   (a) Ignore any compiler errors like. Error: ARCH: Unsupported platform.

   (b) If every thing was done right, a file imgcut.mexw64 is created in the source directory.

2. ***In*** `Ex02.m` ***the input is read from the* DIBCO *folder and the output is written to* GCuts *folder. Compare the results with the output from Otsu binarization and report any improvements. Analyse the code in*** `Ex02.m` ***to know how binarization was performed and explain the steps involved.*** Hint: ***Displaying the images in each step can help.***

3. ***Run*** `Ex02.m` ***on the file provide in*** `SetSourceSink` ***folder and see the ouput of the graph-cut. Try changing the source-sink pixels so that*** (a) ***The bush in red is classified as background pixels.*** (b) ***The trees in blue are classified as background pixels.*** (c) ***The trees and the bush are classified as background pixels. (A background pixel is the white pixel in this case.)***

1.3 Improving the source and sink estimates in Graph cut

In *Ex03.m* the input is read from the *DIBCO* folder and the output is written to *SourceSink* folder. The pixels that can act as possible sources and sinks for a Graph-cut are read from the *Clusters* folder.

4. ***Run* Ex03.m *and report any improvements in the binarized images over the output from simple graph-cut in the previous exercise.***

2 Motion Tracking

Motion analysis is similar to image registration since estimation of velocity could be compared to estimation of disparity. However, in motion analysis we usually have:

- A temporally ordered sequence of images
- These images have the same modality
- Often, two consecutive images are similar

For this assignment you will be working with an image sequence depicting prawns moving around in doughnut shaped container. The image sequence is taken from the dataset of an ongoing research project where the movement of the prawns is studied.

Change directory to `Motion`.

The image sequence you are going to analyse, named `source_sequence.avi`, consists of 581 frames. The first 250 frames could be considered as "easier" whereas the remaining frames contain a number of difficulties. To help you a little a mask (named `bwmask.png`) has been supplied that can be used to remove the central part of the frames if needed. An example result has also been supplied, named `example.avi`.

1. ***Create an algorithm that is able to track the prawns for the first 250 frames. For the report specify how you were able to solve the problem. Also include the relevant parts of the code in the report.*** Hint: ***Take a look at some of the suggested functions below.***

2. ***Now run your algorithm on the final frames of the sequence. Does it fail in any way? If it fails what is the problem? Write a short summary of the problems that arise in the second part of the sequence and suggest some possible solutions (you do not need to implement them).***

2.1 Some help

Some useful functions and code snippets (see the help documentation for each function to see what it does):

- `VideoReader`
- `imextendedmin`
- `regionprops`
- Code snippet for saving an image sequence similar to `example.avi`:

```
aviobj = VideoWriter('Test.avi');
aviobj.Quality = 100;
aviobj.FrameRate = fps;
open(aviobj);
%begin loop
fig = figure(10);
imh = imshow(frame);
hold on;
line(x,y,'marker','o','markersize',10,'LineWidth',2);
F = getframe(fig);
writeVideo(aviobj,F);
%end loop
aviobj = close(aviobj);
```

References

[1] Nicholas R. Howe. *A Laplacian Energy for Document Binarization.* ICDAR, 6–10, 2011.