Exercises – CVML Summer School Grenoble 2010

Comment: *For task 1 and 2 you have to compile some mex files. If you have big trouble compiling any of the mex files then please go to the task 3 which can be done on paper. Also, don't worry if you do not get all done, you can finish the exercises later.*

Homework: *Homework is marked in red. You can look into the homework during the exercise, but then you will very likely not get a chance to look at all exercises.*

Task 1: Experiment with an interactive Image Segmentation System (20+ minutes)

The code is in directory "TaskSegment". First, you have to mexify the following files. Please go into directory: code/Optimization. Then type: "`mex GraphCut.cpp graph.cpp maxflow.cpp`" Then you run the file "main.mat". You will hopefully see several windows popping up. The file does the following. It loads an image: "data\starfish.bmp" and a user input: "data\strafishUser.bmp" (red-means foreground brush; blue-background brush). Then it uses the user brushes to build the likelihood image. Finally it solves the following energy minimization problem:

$$\mathbf{E: \{0,1\}^n \text{ -> } R}$$

$$\mathbf{E(x) = \sum_{i} \theta_i (x_i,z_i) + \sum_{i,j \in N} \theta_{ij} (x_i,x_j,z\_i,z\_j)}$$

where **x** is the unknown segmentation, and **z** the given image.
Also, the neighbourhood system **N** can be 4- or 8-connected.

$\boldsymbol{\theta_i (x_i,z_i)}$ is the color likelihood, computed via Gaussian Mixture models

$$\boldsymbol{\theta_{ij} (x_i,x_j,zi,zj) = |x_i-x_j|\ [\lambda_1 + \lambda_2\ (-\exp\{-ß||z_i-z_j||_2\})\ ]}$$

$$\mathbf{ß=2(Mean(|zi-zj|)\ )^{-1}}$$

this means $\boldsymbol{\lambda_1}$represnts the standard ising prior, and $\boldsymbol{\lambda_2}$the contrast sensitive term.

Task 1.1 Please examine the different outputs. Please have a quick look through the code to understand roughly the flow. (Note, with "visualizeFinalResults = 0" you can disable all of the intermediate results since you do not really need them all the time)

Task 1.2. Understand what the three different parameters are doing (`Parameters.neighborSystem; Parameters.lambda1; Parameters.lambda2).` Note, $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$ are set to an extreme (which one?) and you have to change them.

a) Switch on the ising prior (`Parameters.lambda1`). Now analyse the difference between changing the neighbourhood system. What do you see? What is the prior on the length of the boundary?

b) Switch on the contrast (edge) sensitive information (`Parameters.lambda2`). Vary the relative weights of lambda1 and lambda2. What do you see? Does the effect make sense?

Task 1.3 Try now the user input: “`starfishUser2.bmp`“, can you achieve a perfect result with some parameter settings?

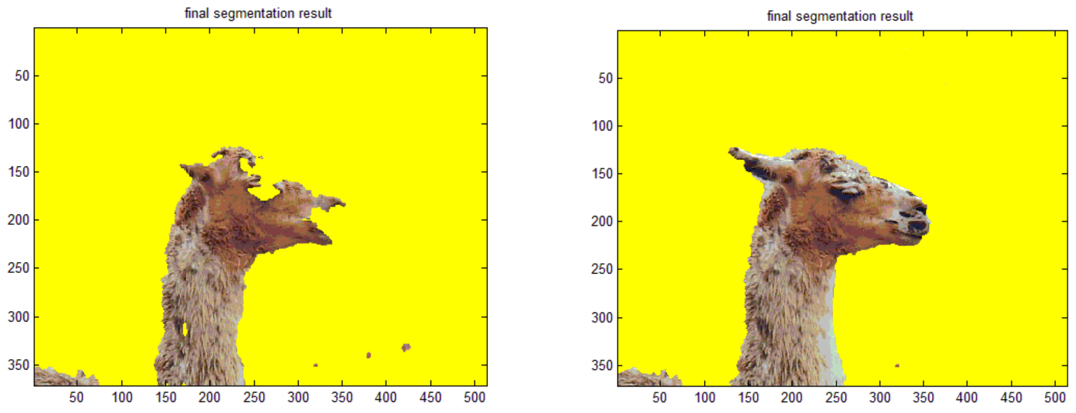
Homework: try different images (or your own). You find more in “..\data”. Please use a graphics software(Gimp is installed) to draw brush strokes. Make sure that a foreground brush is (255,0,0) and a background brush is (0,0,255). How many brushes do you need to get perfect results?

Task 1.4 (do it in the end, or Homework)

In the slides, I have explained that minimizing over both the segmentation and the color models does typically improve the results. So far, the code has used fixed color models derived from the brush strokes only. I would like that you add some lines of code which optimizes both color models and segmentation. For this please do:

1. Copy “main.m” and rename it to “mainIter.m”. Use the example: “`llama.bmp`“
2. Please add code which iterates (e.g. 5times) the part of the code which computes GMMs and the segmentation.
3. About the iterations. In the first iteration you only have the brush strokes as training data. In the second (and all subsequent ones) you have the previous segmentation as training data. This means all pixels in the segmentation which are background are used to train the Background GMM (Gaussian Mixture Model), and all foreground pixels are used to train the foreground GMM.
4. Examine what happens to the segmentation and to the energy in every iteration. Does it improve?

Hopefully you will see something like: before iteration (left) and after iteration (right)



Task 1.5 (Homework) Make a copy of the file: “GraphCut.cpp”. Now rename the file: “GraphCutToFillin.cpp” to “GraphCut.cpp”. Now you open “GraphCut.cpp”. Please understand the file and fill in the 4 lines, which are marked as:
“**** FILL IN *****”. Look at graph.h for an explanation about these functions. Mexify the function with: “`mex GraphCut.cpp graph.cpp maxflow.cpp`”

You can compare your completed file with the original one.

Note graph cut is call in main.m as:

```
LabelOut =
GraphCut(UnaryPotentials,Parameters.lambda1,Parameters.lambda2,0,Parameters
.neighborSystem,im,beta);
```

Task 2: Evaluate different objective functions for Image de-noising (10+ min)

The code is in directory “TaskDenoise”. First please mexfiy energy_min.cpp with: “`mex energy_min.cpp LinkedBlockList.cpp diffusion.cpp`“.
I have provided you with a MATLAB script (denoise_image.m) that tries to denoise a small gray scale image to which noise has been added. There are only 7 possible intensity levels (labels) the different pixels (random variables) can take.

The intensity levels are: L = `[47 65 83 118 136 154 207]`

The script formulates the denoising problem using a pairwise energy function:
E(**x**) = \sum_{i in V} U_i(x_i) + \sum{ij in E} P(x_i,x_j)

The unary terms encourage labels whose intensity is similar to the observed intensity of the image. They are formally defined as:

U_i (x_i) = |I_i – L(x_i) |,

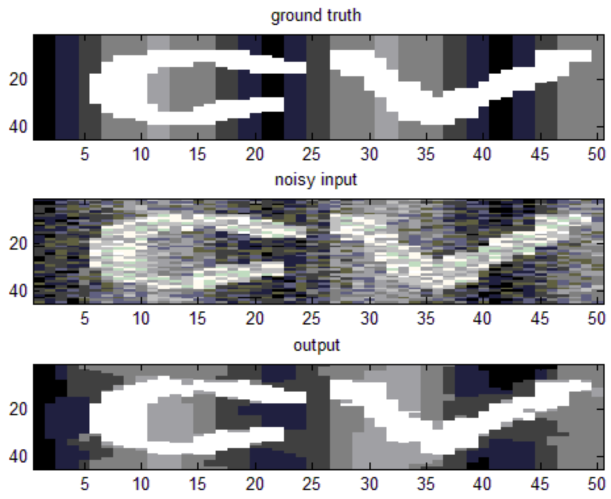
where I_i is the intensity of pixel i in the noisy image, and L(x_i) is the intensity corresponding to label x_i.

The pairwise terms discourage discontinuities in the labels and are formally defined as:

P(x_i,x_j) = Strength * min{|x_i – x_j|, truncation}.

We want you to study the effect these two parameters have on the solution. What is the effect of doing truncation?

For instance, for setting: strength = 10, and truncation = 100, you should get the following result:


ground truth

noisy input

output

Task 3: Some questions for paper and pencil (15+ min)

Please fill in the answers to the following questions. When you are ready discuss with your neighbour. Note, the tutorial slides are online (hopefully) and might help to answer the questions.

Task 3.1

Which of the following pseudo-boolean functions are equivalent to $E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$

A) $8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1 a_2$

B) $6 + 3\bar{a}_1 + 3a_2 + 2a_1 + 3\bar{a}_1 a_2$

C) $8 + \bar{a}_1 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$

Remember: $a_i \in \{0,1\}$, and $\bar{a} = 1-a$

Task 3.2

Which energy function defined over a grid (i.e. 4-neighborhood system) can be minimized exactly in polynomial time for the following pair-wise potentials **g**?
Note, it is $x_i \in \{0,1,\ldots,K\}$, and $K>0$, $T\geq 0$ are arbitrary.

A) $g(x_i,x_j) = (-2)\, |x_i - x_j|$

B) $g(x_i,x_j) = |x_i - x_j|^2$

C) $g(x_i,x_j) = |x_i - x_j|^{0.5}$

D) $g(x_i,x_j) = \min\{ |x_i - x_j| , T\}$

Hint: look at the slides.

Task 3.3a (Homework)

: Reme`mber that a pseudo-boolean function can be optimized with graph cut if the submodularity condition holds, i.e. $f(A) + f(B) \geq f(A \vee B) + f(A \wedge B)$ for all $A,B \in \{0,1\}^n$. Please derive all the non-trivial conditions for 2-cliques and 3-cliques which have to hold.

Hint: one example condition for a 2-clique has been on the slides:

$$f([1,0]) + f([0,1]) \geq f([1,1]) + f([0,0])$$

Task 3.3b (Homework): using the derived conditions please check which of the following potentials can be optimized with graph cut.

A) 2-clique function g(x1,x2)

X_1	X_2	g
0	0	10
0	1	15
1	0	16
1	1	20

B) 2-clique function g(x1,x2)

X_1	X_2	g
0	0	-15
0	1	15
1	0	-16
1	1	15

C) 3-clique function:

X_1	X_2	X_3	g
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	2
1	1	1	1

Task 3.4 (Homework - tricky) Which of the following pair-wise potentials are metric, i.e. can be optimized with the alpha-expansion algorithm?

A) $g(x_i,x_j) = |x_i - x_j|$

B) $g(x_i,x_j) = \min \{ |x_i - x_j|^2 , T\}$

C) $g(x_i,x_j) = \min \{ |x_i - x_j|^{0.5}, T\}$

D) $g(x_i,x_j) = \min \{ |x_i - x_j| , T\}$

Remember: for metric the triangular inequality condition has to hold: $g(x, z) \leq g(x, y) + g(y, z)$

Hints: 1. Think about counter examples.

2. first prove the case where $x<y<z$. Then proof the case where e.g. $y<x<z$.