PROGRAMMING ASSIGNMENT 2

Due: April 16, 2014 (13:30pm)

Investigating Top-Down Influences in Image Segmentation

The process of segmenting an input image refers to the task of determining coherent image regions which are likely to correspond to the existing objects in the image. An image segmentation algorithm follows some basic grouping rules and usually attaches a label to each pixel, denoting which region the pixel belongs to. One such way to model segmentation as a labeling problem is to consider an image as a graph in which the nodes correspond to image pixels and edges reflect pairwise similarities between the pixels. Then the labeling process can be simply formulated as a graph partitioning problem where each resulting subgraph denote an image segment. However, it should be noted that this formulation is a purely bottom-up segmentation model. If we think of natural images, achieving a perfect segmentation is quite hard simply because several ambiguities might be present in the image so a purely bottom-up segmentation methods is doomed to fail. To alleviate this problem, it is very important to combine bottom-up cues with top-down information \cite{1,2,3} to resolve the ambiguities.

The goal of this assignment is to investigate top-down influences in image segmentation. For that purpose, you will experience with both *Normalized Cut*\cite{5}, a well-known (bottom-up) graph-based segmentation method, and its recently proposed extension, referred to as *Biased Normalized Cut*\cite{4}, which incorporates top-down information.

![Biased normalized cuts](image)

Figure 1: Biased normalized cuts \cite{4}. *Top row:* Input image and the corresponding top 4 eigenvectors. *Bottom row:* Biased normalized cuts for various seed sets (taken from \cite{4}).

\footnote{A fast normalized cuts code implemented by Timothee Cour, Stella Yu and Jainbo Shi is available at \url{http://www.timotheecour.com/software/ncut/ncut.html}}

\footnote{An implementation of the biased normalized cut framework can be downloaded from \url{http://www.cs.berkeley.edu/~smaji/projects/biasedNcuts/} You can also use the starter code provided in the course web page.}
Download the set of images and the corresponding (human annotated) ground-truth data provided in the Berkeley segmentation data set 500 (BSDS500)\(^3\).

1. Choose at least 5 images from BSD500 data set.

2. For each input image,
   - Write a script to determine the best value for the number of segments obtained with the Normalized Cuts algorithm.
     
     You can evaluate the success of your segmentation results against human annotations (ground truth) from the BSD500 data set. For that purpose, you can use either one or all of the image segmentation measures used in\(^4\), namely Boundary Error (BE), Rand Index (RI), Global Consistency Error (GCE), and Variation of Information (VoI). Apart from the Rand Index, the lower the values, the better the segmentation quality.

   - Visualize the segmentation result using the code snippets provided in the starter code.

   - Once you obtain the best segmentation result (according to the quantitative analysis described above), try to obtain a better result by using the Biased Normalized Cuts method.

     You can play with the number of seed pixels, or the parameter $\gamma$ which controls the amount of correlation. As the biased normalized cut vector is a weighted combination of $K$ smallest eigenvectors of the normalized graph Laplacian, the quality of your solution is greatly affected by the number of eigenvectors used.

     NOTE: This time you are not required to perform any quantitative analysis, obtaining a visually better segmentation result is enough!

   - Visualize the influence of top-down information by varying the number of seed pixels, just like in Figure 1.

### Grading

The assignment will be graded out of 100 points:

- 0 (no submission), 20 (an attempt at a solution), 40 (a partially correct solution), 60 (a mostly correct solution), 80 (a correct solution), 100 (a particularly creative or insightful solution)

Note: Preparing good report is important as well as your solutions!

---

\(^3\)You can download these from [http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/BSR/BSR_bsd500.tgz](http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/BSR/BSR_bsd500.tgz)

\(^4\)The MATLAB source codes of these four standard image segmentation indices can be downloaded from [http://www.eecs.berkeley.edu/~yang/software/lossy_segmentation/](http://www.eecs.berkeley.edu/~yang/software/lossy_segmentation/)
What to Hand In

You are required to submit all your report along with a short webpage in HTML. For that purpose, prepare a folder containing

- HTML/README.txt (text file containing details about your project)
- HTML/code/ (directory containing all your code)
- HTML/ (directory containing all your documents, including your images)
- HTML/data/ (including your data images)
- HTML/result/ (including your result images)
- HTML/index.html (html report)

Archive this folder as pset1.zip and email to my email address (aykut@cs.hacettepe.edu.tr).

Each student must individually do the coding and prepare detailed HTML report which contains a brief overview of the problems, details of your implementation and results with your observations. If your implementation failed to give a satisfactory results, provide a brief explanation of the reason(s).

Late Policy

You may use up to five extension days (in total) over the course of the semester for the three PSets. Any additional unapproved late submission will be weighted by 0.5.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else’s work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References


