CS 7960, Advanced Image Processing
Spring 2010, Prof. Guido Gerig

Project 6: Normalized Graph Cuts - Graph-based Image Segmentation

Out:	Thursday April-15-2010
Due:	Thursday April-29-2010
Office hours:	Tue 1pm to 3pm, please contact me in advance or ask for other arrangements.

Recommended Readings: Papers and Materials from Lecture on Graph Cuts, in particular **s pages extracted pages from the Trucco and Verri Computer Vision book, see project web-page "Normalized-Graph-Cuts-Intro-GG.pdf"**, papers by Shi and Malik).

Normalized Graph Cuts

Graph cuts and its extended version "normalized graph cuts NGC" have been developed as techniques for clustering. Graphs representing connections between each node to every other nodes, and links between nodes attributed by an affinity measure, are represented as affinity matrices. Eigenvalue/eigenvector analysis applied to these affinity matrices result in eigenvectors which cluster graph nodes with strong affinity. Affinity can measure proximity (distance), intensity differences, color or texture differences, motion and many more. Several affinity measures can be combined into a scalar value by multiplication.

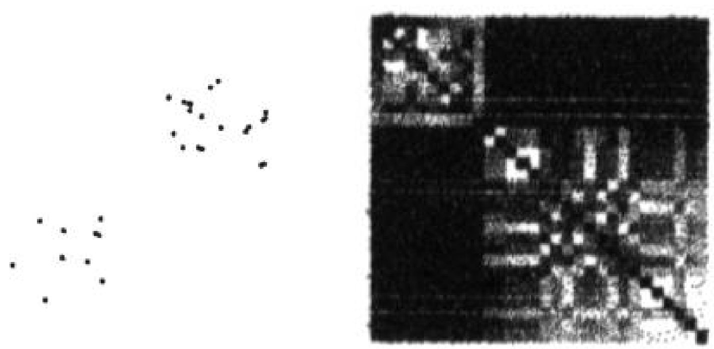

Figure 1: Clusters (left) and affinity matrix representing exponential distance between nodes, shown after matrix is regrouped after eigenvalue analysis (i.e. first 10 rows and columns include the lower left cluster, the reminder the second).

Application to image segmentation is straighforward. Each pixel of an n*m image is defined as a node, and the affinity matrix (n*m x n*m) describes affinity between each pair of pixels. Please note that the initial affinity matrix does not represent any image structure, i.e. does not show image regions of homogeneous properties. The matrix just shows pairwise affinity, and if re-sorted, shows blocks representing pixels with similar affinity. The eigenvalue analysis "reshuffles" rows and columns, i.e. the eigenvector groups the nodes with high affinity.

Instructions

We will follow very closely the book chapter of Trucco and Verri, chapter 16, pages 16 to 34 (see lecture notes). Careful reading is essential to solve this project.
Basically, you need to implement a Graph-Cut (GC) and Normalized-Graph-Cut (NGL) algorithm and apply it to simple test images to understand its pro's, con's and potential merits to real world segmentation problems via experiments. Based on this experience and parameter exploration, you might apply the method to real image(s) of your choice to judge its performance.

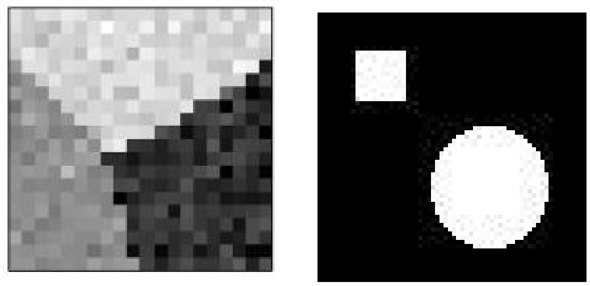

Figure 2: Left: Typical synthetic test image (Shi and Malik, 2000). Right: Example image with disjoint regions (you need to add noise!).

1. **Graph Cut Algorithm:**

   (a) Implement a GC algorithm following chapter 16.4. The algorithm should include the "Affinity by Distance" and "Affinity by Intensity" measures, as described on pages 452 and 453. Based on a digital image of size nxm, an affinity matrix $\mathcal{A}$ is constructed, with dimensions n*m x n*m, i.e. all pixels in rows and columns. Each element of this matrix $\mathcal{A}$ is the affinity $a(i,j)$ between pixels i and j. Please note that the measures are symmetric, and according to the paper the measure for (i,j) and (j,i) are represented as half in each ssociated aelement. Several affinity measures can be combined by multiplication, and the affinity measures themselves include a scaling parameter σ_x to adjust the weighting functions. The eigenvalue problem, $Aw = \lambda w$, is then solved. The resulting eigenvectors will reflect the partition as shown in our document.

   (b) Create simple, small test images, e.g. 10x10, with regions of different gray levels corrupted by noise (see e.g. the test image in Fig. 7 in the Shi and Malik PAMI 2000 paper). The initial image might just show two different regions for a 2-category segmentation. A second image could include one homogeneous region as background, and 2-3 disjoint regions sharing the same intensity (e.g. several blobs with same intensity embedded into a background).

   (c) Test your GC algorithm on the test images, using the intensity affinity measure, the distance affinity measure, and the combination of both. Adjust the σ_x parameters to create optimal results. Display the affinity matrix, it should show a scambled picture since rows and columns are not yet sorted. Display the eigenvectors, as shown in figures 16.20 and 16.21, to judge if they would result in a meaningful decomposition. Find a method to re-arrange the columns and rows to cluster pixels of similar affinity into blocks as shown in figure 16.19.

(d) Judge if you see the problem as discussed in figure 16.23, i.e. that the GC can also result in solutions which are linear combination of eigenvectors associated with "good" clusters.

2. **Graph Cut Algorithm:**
   (a) Implement the NGC procedure as discussed in 16.4.5., following page 460 and parts of the Shi and Malik papers for your implementation.
   (b) Perform tests with the NGC algorithm similarly to the tests with the GC algorithm. Compare results on your synthetic test images.
3. **Eventually run algorithms on real images:**
   (a) Choose images of your choice and run the GC and/or NGC algorithm(s).

Report

You should write up a report summarizing your procedure and discussing your results and experiments. The report should be **written in html** and **accessible to the instructor via a web-browser**, if a web-system is not available you can create a pdf file.

- Short description of how you implemented the affinity matrix and GC and NGC algorithms.
- Describe the success of segmentations using your test images, testing the affinity and combination of affinity measures with their parameters.
- Critical discussion of your results and of your experience with the algorithm, suggestions for changes, etc.