CSC 249/449 Machine Vision: Homework 2
Term: Spring 2018
Instructor: Dr. Chenliang Xu
Due Date: April 3, 2018 11:59PM (Eastern Time)
Version: 1.0 (March 6, 2018)

Constraints: This assignment may be discussed with other students in the class but must be written independently. Over-the-shoulder Matlab coding is strictly prohibited. Web/Google-searching for background material is permitted, but web/google-searching for specific answers to the questions is prohibited (the lectures and readings provide nearly all of the information needed to answer these questions).

Data: You need to download the supplemental data to complete this assignment. All paths in the assignment assume the data is off the local directory. The download has data and code. Note that the code assumes you have a directory structure as follows.
`ROOT/`
`ROOT/one_m/`
`ROOT/two_m/`
`ROOT` is just the top-level directory for your assignments. This structure is needed since some of the files in `two_m` reference those in `one_m`, because we are building on top of each assignment.

Points: This homework assignment totals 80 points.

Problem 1 (20): Affine Photometric Invariance of Harris Operator

One practical goal toward photometric invariance is to seek an operator that is invariant to an affine intensity transformation (spatial range map of size one pixel):

$$\mathbf{I}(x,y)' = a\mathbf{I}(x,y) + b \quad \forall \ pixels(x,y) \in \Lambda \tag{1}$$

where $a, b \in \mathbb{R}$.

For the corner detector operator we discussed in class, $min(eigs(S(\mathbf{I}(W))))$, where $S(\cdot)$ is the structure tensor operating on some window W into image $\mathbf{I}$:

$$S(\mathbf{I}(W)) \doteq \sum_{x,y\in W} \begin{bmatrix} \mathbf{I}_x(x,y)^2 & \mathbf{I}_x(x,y)\mathbf{I}_y(x,y) \\ \mathbf{I}_x(x,y)\mathbf{I}_y(x,y) & \mathbf{I}_y(x,y)^2 \end{bmatrix} \ . \tag{2}$$

Derive and discuss how the $min(eigs(S(\mathbf{I}(W))))$ corner detector operator response on $\mathbf{I}$ is related to that of $\mathbf{I}'$. (Derive the response of $\mathbf{I}'$ is a function of the response of $\mathbf{I}$ and (a,b).)

Problem 2 (60): Spot-It 2: Segmentation

This problem will continue your efforts in MATLAB to implement a computer vision system that can play Spot-It. Here, we react to the limited ability of our interest points to match the cartoon-like drawings on the playing cards. In this assignment, we will compute an over-segmentation of the images, describe each segment using color, and then attempt to extract a full object region using graph cuts.

Data and code: In this version of the assignment, you will work with the same two datasets from the first assignment. Because you are in a different folder, you need to ensure that the code from assignment one is available (`addpath(fullpath('..','one_m'))`). But, the load scripts will expect the data to be in the local directly. So, you can either add a symbolic link on the filesystem (`ln -s ../one_m/cards .`) or change paths in MATLAB just to load the images and then change back.

Work: You will implement a basic superpixel segmentation method, color histogram features, and then create a graph based on the image before executing a provided graph cuts method. You can see a full example of steps in the code in file `twoexample.m`.

1. **(25)** Implement the missing body of code in `slic.m`, which is the main algorithm loop for the SLIC superpixel segmentation algorithm. Use the included paper by Achanta et al. as the reference for the algorithm. The initialization of variables and the recentering step are included for you in the file (to facilitate interaction with the other source files in this assignment). Use the distance function defined in the paper (Eq. 2 or 3 are equivalent); we have set a value of `m` in the code. Do not change `m`. After loading the images (`oneload`), you can invoke the segmentation by `[S,C] = twosegment(1);` where `S` is the segment image and `C` is a structure array containing information about the segments (superpixels); `1` is just the image index. You can visualize them with `twoshow(1,C,S)`.
2. **(10)** Implement the missing body of code in `histvec.m`, which creates a color histogram feature vector. Follow the comments in the code for the details. You call this with `C = tworeduce(1,C,S)`.
3. **(15)** Implement the missing body of code in `twocut.m`, which creates the graph by defining the capacity matrix. See the comments and refer to lecture slides for details. Run this with `B = twocut(S,C,key,tau)`, where `key` is the index to one of the segments in `S` and `C` that you will define what foreground we want, and `tau` is a parameter in $[0, 1]$ that weights between the two types of edges: those linked to source/sink and those linked to neighbors. Visualize results with `twoshow(1,C,S,B)`.
4. **(10)** Run `twoexample.m` to run throught a full example. Change the image a few times; change the key segment a few times; change `tau` a few times. In a text file named 'explain.txt' (less than 2000 characters) explain what you see in the segmentations (both the superpixel oversegmentation and the two-class foreground-background segmentation). Would these be useful in playing Spot-It automatically with a computer vision system?

Example outputs from `twoexample.m` are included in the zip for your help and guidance.

Submission Process: Scan your solutions to Problems 1, and upload it along with Problem 2 on Blackboard. Zip up your MATLAB code and update to the submission folder in the Blackboard assignments. The zip of your folder should expand to one-level up from the main files: the download of the files expanded to `two_m/xyz.m`, and your implementation should do the same.

Be sure to include all of the files, we will run your code. The MATLAB code needs to run on our machines. Do not include any local paths or local files. If any are used, then be sure to upload them with the other code (none are expected). Do not change output formats/locations of the scripts.

Grading and Evaluation: The credit for each problem in this set is given in parentheses at the stated question (sub-question fraction of points is also given at the sub-questions). Partial credit will be given for both paper and MATLAB questions. For MATLAB questions, if the code does not complete, then limited or no credit will be given.