EECS 504 Foundations of Computer Vision: Challenge Project

Term: Fall 2016
Instructor: Jason J. Corso, EECS, University of Michigan
Due Date: 12/13 6AM Eastern Time

Constraints: This is a group assignment. Students may work alone or in teams of up to 3. **Each individual student must submit the full assignment and the writeup must be clear about the students involved in the group.** Web/Google-searching for background material is permitted. However, everything you need to solve these equations is presented in the course notes and background materials, which have been provided already. You may not search explicitly for a problem on the web.

Goals: Test the foundation built up through the semester with an open-ended challenge problem.

Data: Download `cp_data.zip` from canvas. Note that a second download will be available a few days after the assignment is given out. It will contain routines to let you assess your work.

This challenge project has you implementing a computer vision system to play the game Spot-It.

Spot-It: the game has a set of playing cards with a small number of pictures on each. The pictures are simple drawings/icons such as a bottle, a whale, etc. The set of pictures are the same across the whole deck of cards except that they have been translated, rotated and scaled. The cards are organized such that on each pair of cards there is one and only one matching picture identity (differing by translation, scale and rotation).

Data: you have been provided with two datasets: cards and images. Both have five images named `001.png` to `005.png`, and are separate instances of the Spot-It game. Note that in the cards case the green background is not a match.

Work: you need to implement a computer vision system that will find the matching objects across each pair of cards. You may engineer any computer vision system you like, but do thoroughly study the two different datasets (what works on one may not work on another). You may use any of the code you worked on or were provided by me during the semester.

Output: for both datasets, you need to generate full outputs. There are 10 pairs of cards in each set. For example, `001.png` to `002.png` and `001.png` to `005.png`. For each such pair there is one and only one matching object. You need to generate a black-and-white image for each such pair that indicates the match. The image should correspond to the lower-numbered image of each pair (e.g., for `001.png` to `002.png`, it corresponds to `001.png`). You should make the image pixel values be black or 0 everywhere except for at least one pixel that overlays the object that is the matching object, which you should set to white or 255. So, for each pair of images, you need generate a new image that *points to* where the matching object is. Obviously, if you randomly set the white pixel, you will fail to find adequate matches.

You should generate a full set of output images according to the following structure. Create separate sub-directories for `images_results` and `cards_results`. In each, create the output black-and-white image named like `XXX_YYY.png` where `XXX` is the lower-numbered image in the pair and `YYY` is the higher-number image in the pair. E.g., for cards `002.png` to `005.png`, the corresponding output image is located in `cards_results/002_005.png`.

Report: In a two-pages-or-less writeup, describe the following things.

1. Describe your method and the rationalization for your method.
2. Show a good result and a bad result. Explain why your system performs in this manner.
3. Explain any deviation in performance between the cards and the images.

Grading: Your overall performance is not the key here. However, a system that performs very poorly will imply that you did not make a significant effort to engineer a good solution to the problem. We allocate 10 points for the code, 10 points for the performance (as outputted by the to-be-provided code) and 10 points for the writeup.

Submission Process: Submit the pdf, txt and the dot-m files to canvas. Submit the files separately. The txt file will be generated directly the code provided in a few days.

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 run, then limited or no credit will be given.