Computer Vision cs682 **Due: Mar. 18, 2019**
Spring 2019
Dr. Duric

Homework #3 (50p)

In your third homework you will build on the results of Homework #2. You will use edge histograms to compare the gray/color frames of the sequence.
There are several stages to this homework:

1. (10p) You will compute the gray-level edges and color edges for the images used in Homework #2. For gray edges you should convert all color images to gray and compute image gradients. You can use Canny edge detector to select edge points (includes smoothing, derivative computation, hysteresis thresholding and non-maximum suppression). Whatever parameters you choose they should be used an all images. Note that for the next step you need the gradients of the selected points.

   To compute the color gradients you should split the images into r, g, b fields and compute $r_x, g_x, b_x, r_y, g_y, b_y$ components of the gradients using the same process you used for the gray images. Note that it may be appropriate to use lower threshold for the gradients of the color planes than those used for the gray images.

2. (10p) Build gray edge histograms for all images in the sequence. Edge histograms should be 36-bin: you can turn a gradient angle into in index by dividing the orientation angle by 10 and rounding to the nearest integer. You should decide whether you want to use the edge magnitude in the histograms or count all selected edges using the same value, say 1.

3. (20p) To build color edge histograms you should use the gradients computed for all three color planes. You can use the vector $\vec{u} = (r_x + g_x + b_x, r_y + g_y + b_y)$ to determine edge orientation and $m = |r_x| + |g_x| + |b_x| + |r_y| + |g_y| + |b_y|$ for its magnitude. Note that you can proceed to build the color histograms using $\vec{u}, m$ pairs computed for all image points. You may also threshold the remaining points to remove too small values of m.

4. (5p) Write two functions for histograms comparison: histogram intersection and chi-squared measure. Test your functions on the image sequence.

   a) Histogram intersection. Given two color histograms $H_1(\cdot)$ and $H_2(\cdot)$ their intersection is given by
   $$H_1 \cap H_2 = \frac{\sum_i \min\{H_1(i), H_2(i)\}}{\sum_i \max\{H_1(i), H_2(i)\}}.$$
   Large values correspond to high similarity.

   b) Chi-squared measure: χ^2. Given two histograms H_1 and H_2 the χ^2 measure of their similarity is given by
   $$\chi^2(H_1, H_2) = \sum_{i:H_1+H_2>0} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}.$$
   Small values correspond to high similarity.

5. (10p) Using your histogram distance functions compare all image pairs. Create two images displaying your results. Use either colors or gray levels to represent quality of match.

You should illustrate your results on 2 pairs of gray and color images. Show all steps. Make sure that your illustrations of edge orientation and magnitude are meaningful. Hint: you can use heat maps.

Extra credit opportunities:

1. (5p) Use *quiver* function from *matplotlib* to visualize gradients.
2. (5p) The slides illustrating how to compute color gradients using *eigenvalues* and *eigenvectors* are attached. Use the proposed method to obtain the edges.
3. (10p) If you compute eigenvalues and eigenvector using python loops your program will be quite slow. The slides show how it can be done much faster since most of the computation could be done faster using *numpy*. Speed up your code using the method illustrated in the slides.

Submitting your homework

Post the results and your programs/scripts on your webpage; write a report describing your work. Your report must be clear and as brief as possible without compromising comprehension. Post your report on the blackboard. If would be nice if you could use password protection and include the login and the password in your report on the blackboard.