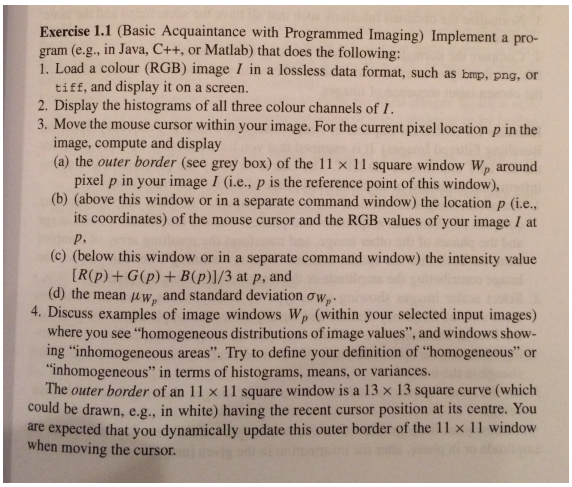
Computer Vision cs682 **Due: Feb. 14, 2020**
Spring 2020
Dr. Duric

Homework #2 (50p)

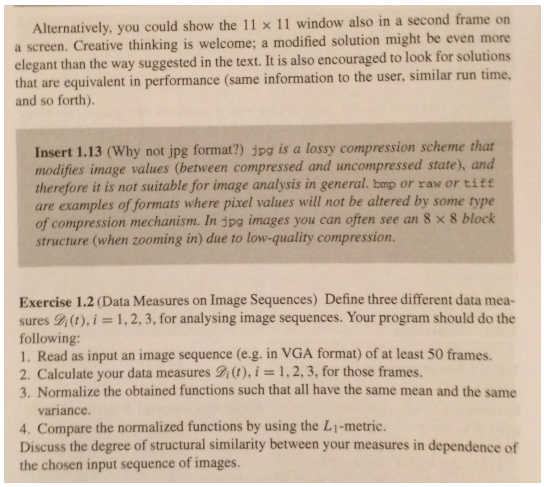
Your second homework is exercise #1.1 from the textbook and modified exercise #1.2 from the book. You will use *OpenCV*. If you do not have the textbook here are copies of the relevant pages.



Exercise 1.1 (Basic Acquaintance with Programmed Imaging) Implement a program (e.g., in Java, C++, or Matlab) that does the following:

1. Load a colour (RGB) image I in a lossless data format, such as `bmp`, `png`, or `tiff`, and display it on a screen.
2. Display the histograms of all three colour channels of I.
3. Move the mouse cursor within your image. For the current pixel location p in the image, compute and display
   (a) the *outer border* (see grey box) of the 11×11 square window W_p around pixel p in your image I (i.e., p is the reference point of this window),
   (b) (above this window or in a separate command window) the location p (i.e., its coordinates) of the mouse cursor and the RGB values of your image I at p,
   (c) (below this window or in a separate command window) the intensity value $[R(p) + G(p) + B(p)]/3$ at p, and
   (d) the mean μ_{W_p} and standard deviation σ_{W_p}.
4. Discuss examples of image windows W_p (within your selected input images) where you see "homogeneous distributions of image values", and windows showing "inhomogeneous areas". Try to define your definition of "homogeneous" or "inhomogeneous" in terms of histograms, means, or variances.

The *outer border* of an 11×11 square window is a 13×13 square curve (which could be drawn, e.g., in white) having the recent cursor position at its centre. You are expected that you dynamically update this outer border of the 11×11 window when moving the cursor.



Alternatively, you could show the 11×11 window also in a second frame on a screen. Creative thinking is welcome; a modified solution might be even more elegant than the way suggested in the text. It is also encouraged to look for solutions that are equivalent in performance (same information to the user, similar run time, and so forth).

Insert 1.13 (Why not jpg format?) *`jpg` is a lossy compression scheme that modifies image values (between compressed and uncompressed state), and therefore it is not suitable for image analysis in general. `bmp` or `raw` or `tiff` are examples of formats where pixel values will not be altered by some type of compression mechanism. In `jpg` images you can often see an 8×8 block structure (when zooming in) due to low-quality compression.*

Exercise 1.2 (Data Measures on Image Sequences) Define three different data measures $\mathcal{D}_i(t)$, $i = 1, 2, 3$, for analysing image sequences. Your program should do the following:

1. Read as input an image sequence (e.g. in VGA format) of at least 50 frames.
2. Calculate your data measures $\mathcal{D}_i(t)$, $i = 1, 2, 3$, for those frames.
3. Normalize the obtained functions such that all have the same mean and the same variance.
4. Compare the normalized functions by using the L_1-metric.

Discuss the degree of structural similarity between your measures in dependence of the chosen input sequence of images.

1. Exercise #1.1 (20p). Use a dialog box to input the image name. Try the exercise with jpeg images too to see the effects of compression. You can find a lot of images at
   `http://cs.gmu.edu/~zduric/cs682/Images/`.

2. (30p) For this part you will use the images from *ST2MainHall4.zip* archive.

   A. (10p) Build color histograms for all images in the sequence. Color histogram should be 512-bin: $[(r/32) * 64 + (g/32) * 8 + b/32]$ will convert a color value into an index. Note that all divisions are integer divisions as in C programming language. That means that they can be accomplished using bit shifts.

   B. (10p) 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.

C. (10p) Using your histogram distance functions compare all image pairs. Create two linearly scaled images displaying your results. The scaling should be between 0 for the minimal distance and $G_{\max}$ for the maximal. Note that the similarity and the distance should have different sense, i.e. the similarity should be minimal when the distance is maximal and vice versa. Use either colors or gray levels to represent quality of match.

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. Your code should be a part of your report, but it also must be downloadable as a *zip* or a *tar* file so that the GTA and I can check it. Post your report with a link to your webpage on blackboard. You *must* use password protection for your web page and put the login and the password into your report.