CS6640 – Project 1
Assigned Sept. 10, 2014
Due Sept. 24 (Just before midnight)
Instructor: Guido Gerig
TAs: Avantika Vardhan, Padmashree Shrinivasa Shetty Teeka,
Office tbd, office hours Mon,Wed 3 - 5pm

Goals

The purpose of this project is for you to develop the software platform you will use for the course (Matlab or C++), to get familiar with the process and expectations for reports, and to write code and experiment with thresholding and connected components in images.

The second purpose of this assignment is to develop your understanding of image histograms, histogram equalization and the use of histograms for image segmentation.

1 Build a Histogram and CDF

Implement a function or routine, for instance in Matlab

```
histogram(I, n, min, max)
```

that takes a 2D array (image) `I` as input (integers or floats) and returns a 1D array of floats (length `n`) that gives the relative frequency of the occurance of greyscale values in each of the `n` bins that equally (to within integer round off) divide the range of (integer) values between `min` and `max` (hint: for 8bit images, just use the range $0 \cdots 255$ for $0 \cdots L-1$). Normalize the histogram by the image size (see course notes) to present a probability density function (pdf).

Build a second function for a cumulative distribution function, e.g. called cdf, that integrates frequencies from the histogram.

```
cdf(histogram, min, max)
```

.

Experiment with these two function on different images and show graphs or plots, in your report, of the images, their histogram and cdf. Explain how the histogram makes sense for each particular image. For the graphs you can use whatever software you want, e.g. Matlab plotting routines, MS Excel, Open office. Document how you did it and show examples.

2 Histogram Equalization

Use the histogram function as implemented above.

1. Implement a function:

   ```
   histoeq(I, n, min, max)
   ```

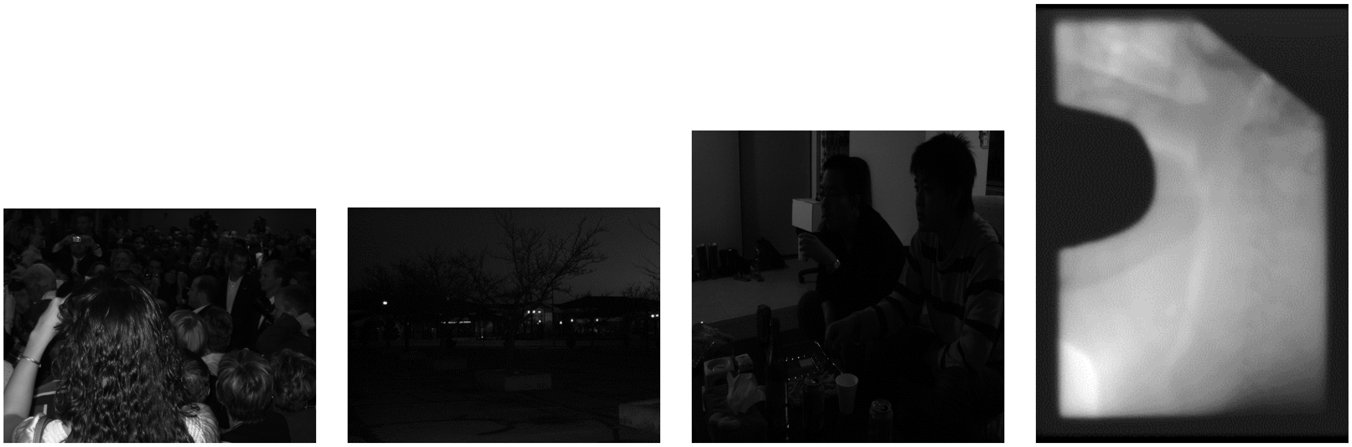

Figure 1: Test images: crowd, university, chang and portal.

that takes a 2D array (or image) `I` as input, calculates normalized histogram and cdf and returns an image of the same type that has undergone histogram equalization using `n` bins and has a range of outputs that spans the range between `min` and `max`. (Please note that hisogram equalization as a code simplifies to mapping the imput intensities through the cdf and multiplyig by (L-1). You may just use the cdf as calculated previously and write a simplified function.).

2. Use this routine to transform and enhance the following images (**??**).

   In addition, include in your discussion 2-3 of your own greyscale images (take own color images, convert to b/w as explained in the MATLAB introduction session) that you think would be interesting from the point of view of this exercise. I.e. that have bad contrast characteristics and that might be improved by histogram equalization.

3. Show graphs of the histograms and of the cdfs of the images before and after equalization and explain their appearance. Did you expect what you see, why and why not?

4. Discuss (briefly) why histogram equalization is or is not effective for each image.

3 Segmentation by Thresholding

As discussed in class, the simplest segmentation is obtained by thresholding of an image if objects and background can be separated. The image histogram may give you a hint on how to choose an optimal threshold but might also be misleading. Given the two figures with checkerboard patterns (Figure **??**), design a method to find the optimal thresholds. Hint: Remember the class' discussion on the mixed Gaussian model.

The following images (see Fig. **??**) are used in this task.

Checkerboard image segmentation

1. Create and display histograms of the two checkerboard figures.

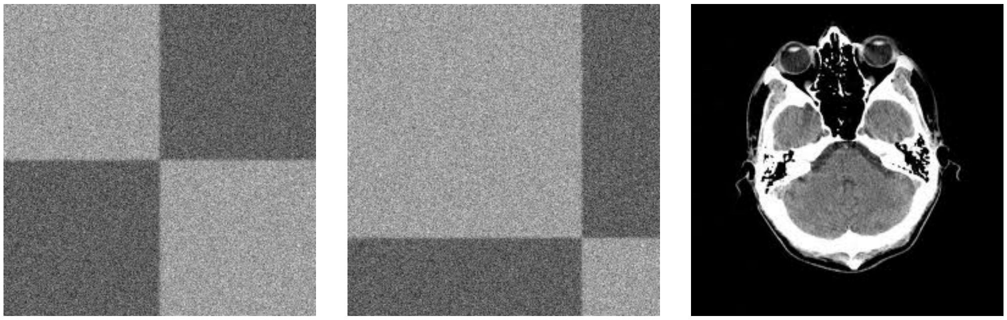

Figure 2: Test images for segmentation

2. Determine a "valley" in the histograms that might indicate the two region categories (e.g. eyballing).
3. Threshold the images at this value, display results and discuss (remember that pixels classified as white in the dark region and vice-versa are errors in the segmentation).
4. Select small regions of interest (e.g. 10x10 patches) in both types of regions and calculate class-specific pdf's for the dark and light gray patches via estimates mean and standard deviation. Compare the results of the pdf's for both images.
5. Design a strategy to select the threshold based on the pdf's for the two categories. Apply this threshold and discuss the resulting images. What is different to above, and why do you expect these differences? You can graph the pdf's by drawing two Gaussians.
6. Given these pdf's and your estimates of the percentage area of the two categories, can you recreate the histogram shapes as actually measured from the original images? (Again, see the course slides discussion segmentation by thresholding).

CT scan segmentation

As a non-expert, please note that the CT image is mainly depicting background (black), soft brain tissue (gray), and bone (white).

1. The reasoning above is often applied to computer tomography image data (see Fig. **??** right), e.g. to separate bone from soft tissue and then display bony structures as 3D graphics to be used for surgical planning.
2. Create the histogram of the CT image. Would the histogram be sufficient to separate bone from non bone?
3. Estimate pdf's of bone and adjacent soft tissue by selection of small test regions and calculation of mean and standard deviations pdf's fit as Gaussian functions (see course slides for a discussion of "supervised segmentation".
4. Determine the best threshold, apply it to the image and discuss.

4 BONUS QUESTION: Mapping to Target Histogram

While histogram equalization optimally stretches the intensity range to min-max, the results may not always convince (your discussion above). Write a function that takes an image with a suboptimal appearance and maps the histogram to an image that looks good to you. E.g, you may create an indoor or outdoor image that is too dark or too bright, then take a second image from a similar viewpoint that looks good, and then adjust the histogram of the poor to the one of the good image.

Follow discussion of histogram matching in the book DIP 3.3.2 and handwritten course notes, where we show that the mapping of input intensities r to output intensities z can be written as $z = G^{-}1(s) = G^{-}1(T(r)$, i.e. a first transform to uniform pdfs via equalization ($T = cdf$) and a second inverse transform G^{1} mapping from uniform back to the target histogram.

```
histomatch(I1, I2, n, min, max)
```

5 Instructions, Requirements, and Restrictions

1. Please use your name **"NAME"** in all reports and submitted materials to uniquely identify your projects.
2. Your report should summarize your approach to solve the problems, show graphs, images of the results and include a critical evaluation/discussion of the results. Please do not copy the project description into your report, just the title is sufficient. The report can be written with any text program of choice, and the handin needs to be in pdf format.
3. Write your project code in a single directory, called `project1-NAME` and the best is to create a compressed tar file of code and images used to run it.
4. We **do not allow to use Matlab toolbox functions** (e.g. Imaging Toolbox) or other existing image processing libraries or solutions found elsewhere. We want you to experience challenges of implementation and give all students the same conditions for code development [1].
5. You should have in your report a short description of each algorithm you used and documentation on how your code is organized. A short summary on most essential core information is sufficient, don't expand too much.
6. Your project report will be in the form of a pdf file.
7. You will submit two tar file created from from your project directory with the unix command *tar -czf project1-NAME.tgz./project1-NAME* to the canvas system, one file being the pdf of he report and the other code and images.
8. **Please remember to look-up the honor code and requirement to provide your own solution as discussed in the syllabus.**
9. **Please look up the late policy as defined in the syllabus**

[1] The core MATLAB package comes with several rudimentary functions that can be used to load, save, and perform custom functions on images.