**Foundations of Computer Vision**

**Project 0**

**Instructor:** Yu Kong

**Due Date**: **11:59PM Feb 25th, 2021**

**Submission**: Upload to MyCourses

**Introduction**

This assignment addresses some of the topics we covered in class. There are four parts to the assignment:

1. Becoming familiar with reading, manipulating, displaying and writing images to ﬁle, as well as using plotting functions. More speciﬁcally the assignment also covers working on various image thresholding methods.

2. Manipulating the pixels of color image and playing with color spaces

3. Extracting features from images.

**Requirements**

You should perform this assignment using Python, OpenCV or Matlab, and it is due on **Tuesday February 25th by 11:59pm**. You are required to submit your code and a report containing short write-ups of what you did along with graphics displaying the results you obtained. You are strongly encouraged to start the assignment early. You are welcome to ask questions and have discussions about the homework on myCourses discussion but please do not post your solutions or any closely related material. If there are parts of the assignment that are not clear to you, or if you come across an error or bug please don’t hesitate to contact the TA or the Instructor. Chances are that other students are also encountering similar issues. According to the RIT academic integrity policy, you are not allowed to share your solutions or post them to any publicly available location.

You are allowed to collaborate with other student(s) in your team. However, you are required to code the solution yourself. Copying others’ code and changing all the variable names is not permitted! For this assignment, you are not allowed to use solutions from similar assignments in courses from other institutions, or those found elsewhere on the web. If you access such solutions YOU MUST refer to them in your submission write-up (penalties may apply). Your solutions should be uploaded to myCourses via Dropbox before the deadline.

Your submitted zipped ﬁle for this assignment should be named **Firstname-Lastname.zip**. Failure to follow this naming convention will result in delays in getting your grade. Your zipped ﬁle should contain: (i) a PDF ﬁle named **Firstname-Lastname**.pdf with your report, showing output images for each part of the assignment and explanatory text, where appropriate; (ii) the source code used to generate the solutions (with code comments). Please make sure your code compiles and runs successfully before submitting it. We should be able to execute your code for each part of the assignment in turn. Include a readme ﬁle if necessary (especially if using several external libraries), to run your code.

Using these images (original attached)



1. **Color Space (10/100)**

Please convert the color space of this given image **Lena.png** from RGB to CIE XYZ using Matlab/Python, and plot your result.

1. **Color histogram (10/100)**

Please compute the color histogram of this given image **Lena.png** using Matlab/Python, and plot your result.

1. **Basic image manipulations (30/100)**

The goal of this problem is to get you to become familiar with basic image manipulations, including reading in images, writing out their modiﬁed versions, generating image histograms and running various types of image thresholding techniques. For this part of the assignment, from the folder images, you are to read in the image **sonnet.png** shown in Figure 1a:

(a) Plot its histogram, as shown in Figure 1b and use this to manually determine a value to use as the global threshold.

(b) Traverse through all the pixels in the image and set the value to 255 if greater than your threshold, else 0. Display your newly modiﬁed black and white image. Include your result in your report as in Figure 1c.

(c) Now, implement adaptive thresholding, a localized version of what you performed above, using any one of the following formulas as we discussed in class. Turn in the version that gives the best looking results on your image. Include your best result in your report as shown in Figure 1d. The three options are:

(i) t = mean(N × N) + C

(ii) t = median(N × N) + C

(iii) t = ((max − min)/2) + C

Note: You will have to ﬁnd the appropriate values of the window size N × N and the constant C. I used trial-and-error but you might be able to ﬁnd a better way. But remember that these values change based on which threshold ﬁnding technique you use.

Your submission should contain the all the code ﬁles used to address this problem. We will run the code on other similar test images. Your report should contain a short writeup to include which threshold ﬁnding approach you used, graphics of the results you obtained, similar to Figure 1 and what values of N and C worked best under those circumstances. Extra credit will be given for any additionally creative solutions and/or any insights shared about how the problem was solved.

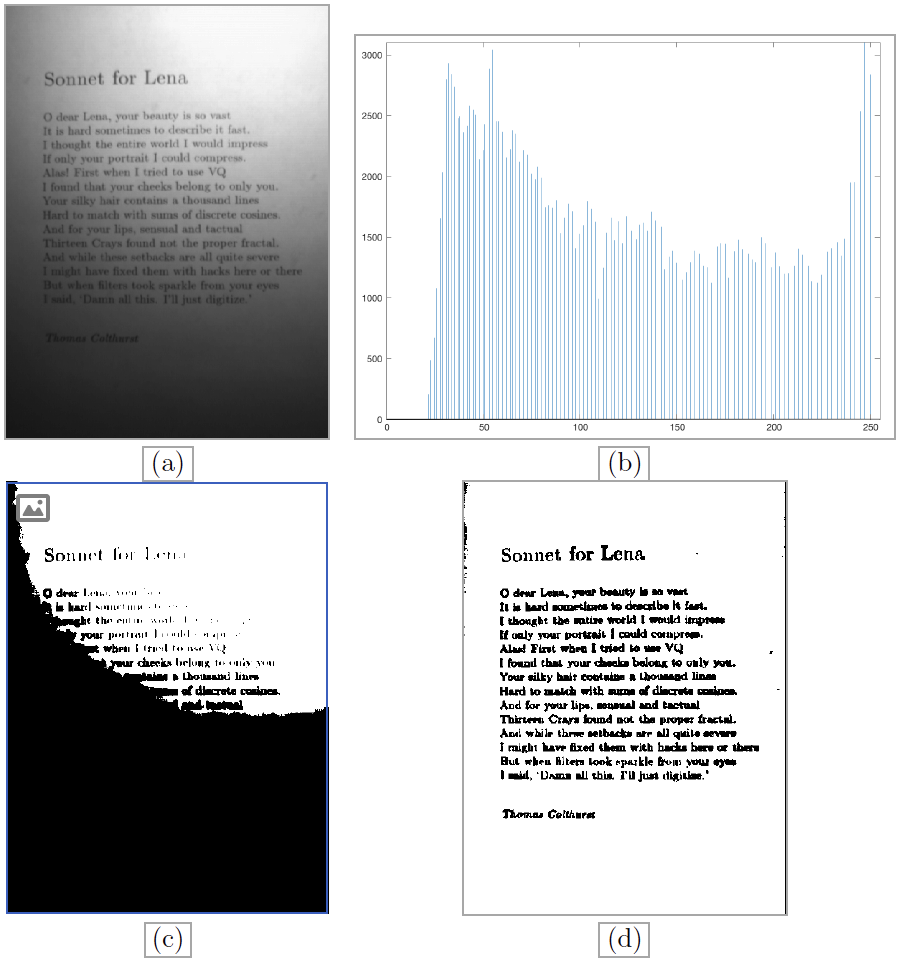


Figure 1: (a) the original image; (b) the histogram of the original; (c) the results after using my best guess at a global threshold value and (d) my best result using adaptive thresholding.

1. **Interest points (50/100)**
2. Use SIFT code at <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html> or <http://www.vlfeat.org/overview/sift.html> to extract SIFT features on data from <https://github.com/kushalvyas/Bag-of-Visual-Words-Python/tree/master/images/train>
3. Visualize SIFT features detected on one representative image of each category.
4. Create your Bag-of-words model, and visualize the histogram of these images (one representative image of each category). Reference can be found at <https://github.com/kushalvyas/Bag-of-Visual-Words-Python>

Please generate a report for your visualization.