Homework 1
Computer Vision, Spring 2019
Due Date: February 6, 2019
Total Points: 20

Note 1: There will be a total of 7 homeworks. All homeworks together will account for 60% of your final grade. Some homeworks are lighter than others; their relative weights in your final homework grade are determined by the total number of points listed at the top.

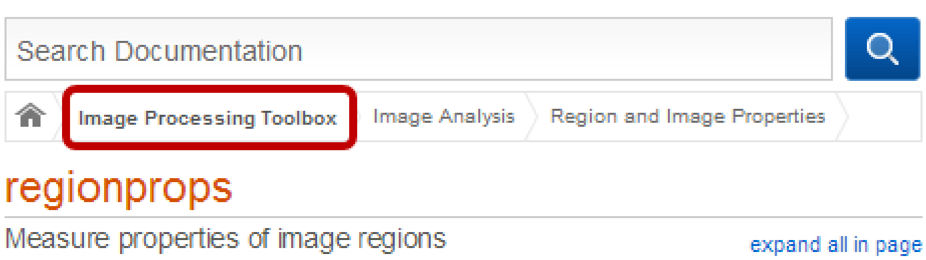
Note 2: There will be three types of problems in homeworks: **written problems**, **programming walkthroughs** and **programming challenges**. A homework may or may not contain all three types of problems. For example, Homework 1 has only written problems and programming walkthroughs. Below is a brief description of each problem type.

Written Problem: These are mainly analytical problems. Written solutions must be submitted *as a hard copy at the beginning of class on the due date*. You will receive credit ONLY if you support your answers with conclusive arguments/sketches/proofs. When in doubt, a mathematical proof is the safest bet.

Programming Walkthrough: These are "lightweight" programming assignments, which essentially walk you through partially complete code and ask you to fill in the missing segments. These assignments are designed with the following two goals: 1) to help you get started with MATLAB (if you are not yet familiar with it), 2) to demonstrate some of the important concepts discussed in class in MATLAB.

Programming Challenge: These are programming challenges to solve a variety of computer vision tasks using MATLAB. In most cases a testing framework or skeleton code will be provided. Your submitted code must work with these.

For all programming walkthroughs and challenges, **you may not use any functions from MATLAB's Image Processing or Computer Vision toolboxes**, except for functions that are explicitly permitted. To check if a particular function is from one of these toolboxes, type `doc function_name` into the Command Window to view the function's documentation. At the very top of the documentation window, the source of the function will be listed. For example:



In addition, the usage of one or more **specific built-in MATLAB commands may be barred**. Special instructions regarding allowed or disallowed functions will be explicitly stated in the description of each programming problem. You are required to submit the completed code and the generated outputs for both programming walkthroughs and challenges. Follow the separate document titled "**Guidelines for Programming Assignments**" for programming guidelines and submission information.

Written Assignments

Problem 1: Consider a pinhole camera with perspective projection.

a. Given a circular disk that lies anywhere on a plane parallel to the image plane, what is the shape of the image of the disk? **(1 point)**

b. Suppose the area of the image of the circular disk is 1 mm^2 when the distance from the pinhole to the plane of the disk is 1 meter. What is the area of the image of the circular disk if the distance is doubled? **(1 point)**

c. Now, replace the disk with a sphere. What is the shape of the image of the sphere? (Be careful with this one.) **(2 points)**

Problem 2: We have discussed the concept of hyperfocal distance in class. Given an imaging system with focal length *f*, f-number *N* and an imaging sensor with pixel size *c*, prove that the hyperfocal distance *H* of the imaging system is

$$H = \frac{f^2}{Nc} + f. \qquad \textbf{(2 points)}$$

Problem 3:

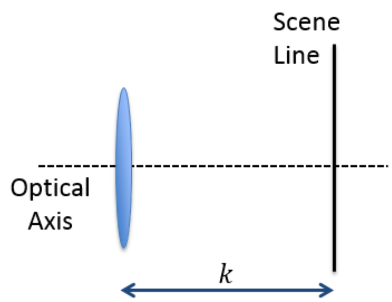


Figure 1

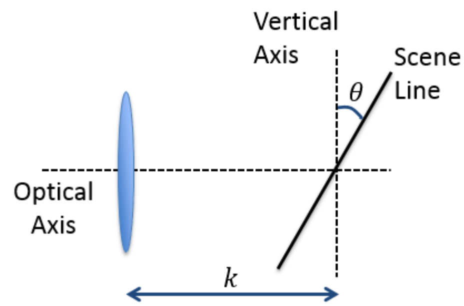


Figure 2

a. Consider the imaging system shown in Figure 1. For this problem assume that the world is 2D, so instead of image and *scene planes*, we have image and *scene lines*. The focal length of the lens is f. Now, consider a scene line that is perpendicular to the optical axis of the lens and at a distance of k from the lens. At what distance from the lens would a focused image of the scene line be formed? **(1 point)**

b. Suppose the scene line is not perpendicular to the optical axis, but makes an angle of θ with the vertical axis (Figure 2). Prove that the image of the scene line is still a line, but it is tilted. **(3 points)**

c. If the image of the tilted scene line is a line making an angle ϕ with the vertical, then show that

$$\tan(\phi) = \frac{f}{k - f}\tan(\theta),$$

where k is the distance between the scene line and the lens along the optical axis. **(3 points)**

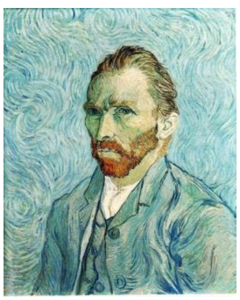
Note: Although you have derived the above equation for a 2D world with *line-scenes*, the equation holds in a 3D world with *planar-scenes*. This is known as *Scheimpflug condition*.

Programming Assignments

The goal of this programming assignment is to get you started with MATLAB for image processing and computer vision. The accompanying MATLAB script **runHw1.m** contains instructions and partially complete code to illustrate some of the basic concepts of MATLAB. Your tasks are to fill in the incomplete code, and to generate the results by executing the script. Include both the completed script and the outputs in your submission.

Walkthrough 1: Go through this brief introduction of MATLAB [1]. Have fun experimenting with different commands. Additional tutorials can be found here [2] [3]. You are not required to submit any code for this Walkthrough 1. **(0 points)**

Walkthrough 2: Fill in the missing parts in **hw1_walkthrough2.m** to read an image and generate a 2x2 collage as shown below. The four patches of the collage are the original image and its red, green and blue channels. Submit both the completed script and the output. **(3 points)**


Input [4]


Output

Walkthrough 3: Complete **hw1_walkthrough3.m** to superimpose the "I Love NY" logo on top of a Manhattan scene. **Image Processing Toolbox functions permitted: im2bw, imresize.** **(4 points)**

Inputs [5] [6]

Output

References and Image Credits

[1] MATLAB Introduction (UCSD). [Online].
http://cseweb.ucsd.edu/~sjb/classes/matlab/matlab.intro.html

[2] MATLAB Documentation Center. [Online].
http://www.mathworks.com/help/documentation-center.html

[3] MATLAB Tutorials. [Online].
http://www.mathworks.com/academia/student_center/tutorials/launchpad.html

[4] [Online]. http://en.wikipedia.org/wiki/File:SelbstPortrait_VG2.jpg

[5] I Love New York logo. [Online].
http://en.wikipedia.org/wiki/File:I_Love_New_York.svg

[6] Ruben Moreno Montoliu. [Online].
http://www.flickr.com/photos/ruben3d/4392232665/