Assignment A1: Image Representation and Formation

CS 4640
Fall 2019

Assigned: 20 August 2019

Due: 5 September 2019 (handin PDF and .m files; turn in hardcopy of PDF printout in class)

1. Do Chapters 1 and 2 exercises. You are expected to know this material, and it may be used on quizzes.

2. The map of Utah (map1.jpg) in the class data directory, explore r,g,b models to segment the semantic components of the map (e.g., water bodies, forests, red roads, etc.). Try to use imapprox. Propose some performance measure and use it in your evaluation. Report what you tried and what results you got.

3. Matlab provides the *rgb2gray* function to convert from rgb images to gray level. Test the hypothesis that *rgb2gray* uses the function given in the book (p. 11, Eqn (1.1)):

$$g = \alpha r + \beta g + \gamma b$$

and if so, what the values of α, β and γ are. To do this use the backslash operator or *lsqlin*. If Matlab uses this approach, then each gray level value gives a linear equation in terms of rgb:

$$g = [\alpha, \beta, \gamma] \cdot [r, g, b]^T$$

Put these into a system and solve. Describe your work, and draw conclusions based on the results.

4. Develop a camera function, *CS4640_camera*, which uses Eqn (2.21) to produce a gray level image from a set of 3D points. Demonstrate it on a variety of sets of points, including

a set which captures the converging railroad in Fig. 2.16 (e.g., points along the rails). Describe development and issues that were dealt with.

For this problem, handin Matlab .m files for the functions described by the header below. Note that one of these is a driver, *CS4640_A1_driver*, which is available in the code directory; it creates inputs for the camera function and runs the function on those inputs to obtain the output.

Note: DO NOT USE SCRIPTS. No function should write to the interpreter, draw, etc.

```
function im = CS4640_camera(f,pts,M,N,S,sigma2)
% CS4640_camera - produce an image from a set of 3D points
% On input:
%     f (float): focal length (assume set to 1)
%     pts (nx3 array): X,Y,Z 3D points from scene
%     M (int): number of rows in output image
%     N (int): number of cols in output image
%     S (int): size of Gaussian filter window (one side)
%     sigma2 (float): variance for Gaussian
% On output:
%     im (MxN array): output image
%       - 3D points at x,y extremes lie on image edges
%       - pixel intensities are scaled by Z value
%       - image is flipped up down (e.g., use flipud)
%       - make sure intensities are between 0 and 255
% Call:
%     im = CS4640_camera(f,pts,M,N,S,sigma2);
% Author:
%     <Your name>
%     UU
%     Fall 2019
%
```

```
function im = CS4640_A1_driver(delx,M,N,S,sigma2)
% CS4640_A1_driver - runs CS4640_camera with railways
% On input:
%     delx (float): step in Z value for line
%     M (int): row dimension for image
%     N (int): col dimension for image
```

```
%     S (int): side length of Gaussian filter matrix
%     sigma2 (float): variance for Gaussian filter
% On output:
%     im (MxN array): output image of railway convergence
% Call:
%     im = CS4640_run_camera(0.01,101,101,15,0.1);
% Author:
%     <Your name>
%     UU
%     Fall 2019
%
```