

## CSE152 – Intro to Computer Vision – Assignment #1

Instructor: Prof. David Kriegman.

<http://cseweb.ucsd.edu/classes/sp12/cse152-a>

Due Date: Thu. April 26, 2012.

### Instructions:

- Attempt all questions
- Turn in a **hard copy in class**, which should include answers to all paper-and-pencil questions and a printout of all matlab code and results for programming questions
- In addition, **email a copy of your code** with the subject *CSE152 Assignment 1* to sbranson@cs.ucsd.edu. Attach code as a *zip* or *tar* file.

### Questions:

1. [35 points] Suppose we are viewing a square with vertices  $P_1 = (-1, -1, 0)^T$ ,  $P_2 = (1, -1, 0)^T$ ,  $P_3 = (1, 1, 0)^T$ ,  $P_4 = (-1, 1, 0)^T$  in world coordinates. Consider a perspective view of this square acquired from pinhole camera  $A$ .
  - (a) [3 points] Assume camera  $A$  has an image plane at a focal length of  $f_A = 1$  in front of the camera. Find the intrinsic parameter matrix  ${}^A M_{in}$  for camera  $A$ .
  - (b) [5 points] Assume the camera is positioned (relative to world coordinates) by applying a rotation of  $\frac{\pi}{4}$  radians about the x-axis, followed by a rotation of  $-\frac{\pi}{4}$  radians about the y-axis. The center of the camera is placed such that the origin is at  ${}^A O = (0, .5, 2)^T$  in the coordinate system of camera  $A$ . Compute the 3X3 rotation matrix  ${}^A R$  going from world coordinates to camera coordinates.
  - (c) [4 points] Compute the 4X4 extrinsic parameter matrix  ${}^A M_{ex}$  defining a rigid transformation going from world coordinates to camera coordinates. Use it to solve for the vertices of the square in camera  $A$ 's coordinate system  ${}^A P_1, {}^A P_2, {}^A P_3, {}^A P_4$ .
  - (d) [5 points] Combine the results of the previous two parts to solve for the 3X4 perspective camera projection matrix. Use it to solve for the projection of  $P_1, P_2, P_3, P_4$  onto the image plane and plot their locations (e.g., draw a quadrilateral depicting the projected image of the square).
  - (e) [4 points] Solve for the 2 vanishing points of the projected image of the square.
  - (f) [4 points] Find the affine projection matrix that approximates the perspective projection from part c and corresponds to an expansion about the center of the square  $O^A$ . Compute the projected locations of  ${}^A P_1, {}^A P_2, {}^A P_3, {}^A P_4$  under this affine camera model, and plot the results as in part d.
  - (g) [4 points] Find the orthographic projection matrix that approximates the perspective projection from part c and corresponds to an expansion about the center of the square  ${}^A O$ . Compute the projected locations of  ${}^A P_1, {}^A P_2, {}^A P_3, {}^A P_4$  under this orthographic camera model, and plot the results as in part d.
  - (h) [6 points] Suppose we were to move the camera to a new configuration  $B$ , obtained by zooming in the camera while panning out, such that the projected image of the square stays roughly the same size. This effect is achieved by setting the focal length of the camera to  $f_B = 10$ , while moving the camera further away to  ${}^B O = (0, 5, 20)^T$ . Are the affine and orthographic camera approximations more appropriate for camera configuration

$A$  or for camera configuration  $B$  (a camera approximation is appropriate if the projection onto the image plane is similar to the projection using the full perspective camera matrix)? Explain your answer.

2. [15 points] Write a matlab function `imgOut = TRANSFORMHSV(imgIn, $\theta$ , $s$ , $v$ )` to manipulate the color of an image in HSV color space. The function should do the following:

- Convert the image from RGB to HSV color space
- Apply a clockwise rotation of the hue channel by  $\theta$  degrees
- Multiply the saturation channel by  $s$
- Multiply the value channel by  $v$
- Convert the transformed HSV image back into RGB color space

DO NOT use the built-in `rgb2hsv` or `hsv2rgb` functions in Matlab, except to compare your results. Test your function on the image `color.bmp`, and generate results for `TRANSFORMHSV(img,90,1,1)`, `TRANSFORMHSV(img,0,.3,1)`, `TRANSFORMHSV(img,0,1,.3)`. Include the generated images in your writeup.

3. [20 points] In this exercise, we will render the image of an ellipsoid illuminated by two different point light sources using a Lambertian reflectance model. The ellipsoid is defined by the equation  $z = \sqrt{x^2 + 4y^2}$  for all  $x^2 + 4y^2 \leq 1$ . The first point light source is located at  $(1, 1, 2)^T$ , and the second light source is located at  $(-1, -1, 2)^T$ . Both light sources have intensity .5. The ellipsoid surface is Lambertian, and has an albedo of 1 throughout the entire surface.

Write a Matlab function that renders a 401x201 image of the ellipsoid with pixels corresponding to the locations  $x = -1 : 0.005 : 1$  and  $y = -.5 : 0.005 : .5$ . Set all pixels outside the ellipsoid  $x^2 + 4y^2 > 1$  equal to 0. Generate an image of the ellipsoid that is illuminated by just the first light source, a second image that is illuminated by just the second light source, and a 3rd image that is illuminated by both light sources. Include all 3 output images in your writeup.