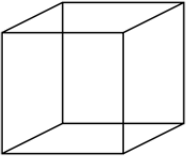


Homework 2

Due Monday, February 4, 2013

Notes: Please email me your solutions for these problems (in order) as a single Word or PDF document. If you do a problem on paper by hand, please scan it in and paste it into the document (although I would prefer it typed!).

- (20 pts) A cube has vertices in world coordinates: $(0,0,0)$, $(1,0,0)$, $(1,1,0)$, $(0,1,0)$, $(0,0,1)$, $(1,0,1)$, $(1,1,1)$, $(0,1,1)$. A camera is located at $(X,Y,Z) = (3,-4,2)$ in world coordinates. The camera points directly at the origin and there is no roll about the axis (i.e., the +Z axis of the world points up in the image). Generate an image of a wireframe model of the cube as if were seen by the camera, as shown in the figure. Assume a pinhole camera model, with focal length = 600 pixels, where the image size is 640 pixels wide by 480 pixels high. Hints:


 - You can draw a line in the image using Matlab's line function.
 - The second vertex, the one with world coordinates $(X,Y,Z) = (1,0,0)$, projects to pixel location $(x,y) = (419, 268)$, rounded to the nearest pixel. Of course, the first vertex should be in the exact middle of the image.
 - A good way to create the rotation matrix ${}^w_c R$ is to use the fact that its columns are the coordinate axes of the camera, expressed in world coordinates (see Lecture 4 slide 5). For example, the 3rd column is the z axis of the camera, expressed in world coordinates. This is just the direction in which the camera is pointing.
- (15 pts) Repeat the previous problem, but this time generate an image of the cube assuming a weak perspective projection model (Lecture 5, slide 8).
- (10 pts) Compute the following quantities by hand for the 5x5 image below. Assume as usual that +x is to the right, and +y is down.

 - What is the gradient at the pixel in at the center? Use the “central difference” operators; i.e., $df/dx \approx [-0.5 \ 0 \ +0.5]$, $df/dy \approx [-0.5 \ 0 \ +0.5]^T$.
 - What is the magnitude of the gradient at that point?
 - What is the direction of the gradient at that point?

6	7	8	9	10
5	6	7	8	9
4	5	6	7	8
3	4	5	6	7
2	3	4	5	6

- (15 pts) Implement a program to correlate a box filter consisting of a 15x15 array of 1's with the “pout.tif” image (Lecture 6, slide 10). For this problem, do not use the Matlab function

`imfilter`, or any related functions such as `convn` in your implementation. Estimate by hand the total number of addition operations that are needed (note – no multiplications are needed).

5. (10 pts) A fast method to implement box filters of various sizes is to create an “integral image”. Create a “summed area table” or “integral image” for the 5x5 image shown below, as illustrated in Figure 3.17 in the textbook. Find the sum of the center 3x3 pixel region directly, and show that you get the same result using equation 3.32.

```

3  4  7  8  7
6  8  3  5  3
6  3  5  2  7
2  5  6  2  2
1  2  8  3  8

```

6. (10 pts) The following is a binary image.

	1	1			1			1	
1	1				1			1	
1		1	1	1	1	1	1		1
1						1	1		
1		1	1	1					
1		1		1		1		1	
1		1		1		1	1	1	
1		1	1	1		1	1	1	
1							1		
1	1	1	1	1	1	1	1		

- (a) Label (by hand) the connected components assuming 4-neighbor connectivity. Compare your result to that obtained by Matlab’s `bwlabel`.
- (b) Label (by hand) the connected components assuming 8-neighbor connectivity. Compare your result to that obtained by Matlab’s `bwlabel`.
7. (20 pts) The images “robot1.jpg”, “robot2.jpg”, and “robot3.jpg” are on the course website. Using the same program (i.e., you can’t change any thresholds or parameters other than the name of the image file), find the concentric circle targets in all three images and draw crosshairs on top of the targets, on the original images. Your program should find all the valid fiducial targets, and reject false targets (you may have to use additional tests to reject false targets). Give your program as well as the images with the overlays. Extra credit: Automatically identify the targets and label them with “UL”, “UM”, “UR”, “LL”, or “LR” for upper left, upper right, lower left, etc.