Assignment 2

Due: May 1

Part I: Written Problems

1. Consider modeling a camera with a pinhole model. Let the image plane be an array of 640 by 480 square pixels that is located 25mm from the center of projection. Let the image plane be 20mm in the horizontal direction.

   a. In degrees, what is the horizontal and vertical field of view? Show your work.

   b. Consider imaging a plane that is known to be located precisely $1m$ from the optical center and parallel to the image plane, and you are told that your edge detector has an accuracy of 0.3 pixel. For an edge detected at the very center of the image, how accurately can you locate the edge in 3D?

   c. If the edge is detected (+200, +200) pixels from the center, how accurately can you locate the edge in 3D? Is it the same as your answer to part b?

2. Given two 1-D signals f(x) and g(x), show that

   $$\frac{d}{dx}[f(x) * g(x)] = \frac{df(x)}{dx} * g(x)$$

   and that

   $$\frac{d}{dx}[f(x) * g(x)] = f(x) * \frac{dg(x)}{dx}$$

   where $*$ denotes convolution.

Part II: Canny Edge Detector

Your goal is to implement the Canny edge detector as described in the text (Trucco and Veri) including the three main steps of filtering, nonmaximum suppression, and hysteresis thresholding and linking. You will apply your edge detector to a set of images found on the web page. You should be able to specify three parameters: σ (the filter standard deviation), τ_l (the low threshold), τ_h (the high threshold).

Some specific notes:

- For convolution, your filter width should be 3σ where σ is measured in pixels.
- You must implement the convolution operation yourself (e.g., you can't use the built-in conv2 or gradient routines in matlab).
- For simplicity, you can ignore the boundary in convolution, i.e if the image size is (H,W) and kernel size is 2m+1, then the result has size(H-2m,W-2m).
- To explore the role of changing each of three parameters, you will include the output of running the edge detector six times. In the first pair, choose reasonable values for τ_l and τ_h, and then vary σ to show the effect of varying filter width. In the second pair, fix σ and τ_l and vary τ_h. For the final pair. fix σ and τ_h and vary τ_l.

Hand in:

1. Source code. Write separate functions for each step, i.e: filtering, nonmaximum suppression, and hysteresis thresholding and linking.
2. Pick one image and one value of σ, show four images with:
   (a) The component of the gradient with respect to x.
   (b) The component of the gradient with respect to y.
   (c) The magnitude of the gradient (edge strength).
   (d) The result of nonmaximum supression.
3. For each of the test images, the result of applying the edge detector with the six choices of parameters as described above.
4. Zip all files and email to TA at d1vu@cs.ucsd.edu. You may convert the images to other format (jpeg, gif etc) to make the file smaller.
5. Also, **print the the source code and images described above and hand in together with the written problems in class, May 1**. Clearly label your figures.

Hint (for Matlab users):

1. You can use Matlab conv2, gradient etc to verify your program.
2. To display image, use (assume data is in *im*):
   (a) **imagesc**(im);
   (b) **axis**('image');
   (c) **colormap**(gray); %*if the image appears not in gray scale.*
3. Use **imwrite** to create the image.
4. Type **help** function_name to view document about the function. Other help and tutorial can found at

   http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml.