EE5178 : Modern Computer Vision

Programming Assignment 2: Canny Edge Detection

Instructions

- For any questions, please schedule a time with TAs before deadline according to their convenience. Please use moodle discussion threads for posting your doubts and also check it before mailing to TAs, if the same question has been asked earlier.
- Submit a single zip file in the moodle named as PA2-Rollno.zip. Submit your detailed answers and your code with relevant output in a Jupyter Notebook. Kindly comment your code generously.
- Read the problem fully to understand the whole procedure.

Implementing Canny Edge Detection

You are required to implement the canny edge detection algorithm as described in class. Please follow the steps mentioned below.

1. Read the image 'clown.jpeg' and display it
2. Convert it into a grayscale image
3. To suppress the noise use a Gaussian kernel to smoothen it. Keep the kernel size as 5 x 5 and sigma = 1.5
4. Apply the standard Sobel operator Gx and Gy discussed in class. Display the filtered outputs. Also show the gradient magnitude and angle images.
5. Apply non-maximum suppression as discussed in class. The exact method requires computing the imaginary pixels shown in gray in the image below using interpolation methods such as bilinear and bicubic interpolation. As a simplification just locate the image pixels which are spatially closest to the imaginary gray pixels and use those pixels as the neighboring pixels for non-maxima suppression. This way you can bypass the interpolation step and have a computationally lighter method if you so desire. Show the Non-Maximum Suppression output.

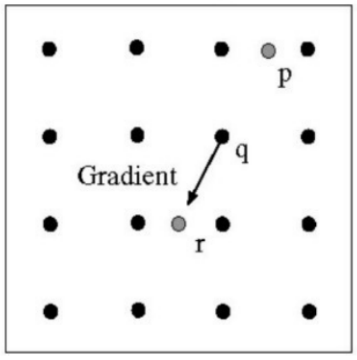

Figure 1: The exact method requires you to do interpolation. As an alternative adopt the simplified approach described in Step 5

6. Instead of using the Double thresholding Hysteresis used in the exact algorithm, simply do a single thresholding operation. Pixels with values less than the threshold should be suppressed. Use the median value of the magnitude image computed in Step 4 as the threshold value. Show the final output.

7. Repeat the above steps (1-6), but increase the sigma of the Gaussian kernel to 3. Choose your filter size appropriately. Display the intermediate results as well. How does the final output differ from that of the previous question?
