Canny Edge Detection

**Problem Statement:**

Implement an edge detection algorithm by following the set of goals for an edge detector and using the optimal method for achieving them as defined by John Canny

**Implementation:**

Language used – Python 3+

Libraries used – NumPy, OpenCV, Math

The following process is followed:

1. Read a gray scale image and store it as a matrix named I
2. Create a one-dimensional Gaussian mask G to convolve with I. The standard deviation(s) of this Gaussian is a parameter to the edge detector (call it σ > 0)
3. Create a one-dimensional mask for the first derivative of the Gaussian in the x and y directions; call these Gx and Gy. The same σ>0 value is used as in step 2.
4. Convolve Ix with Gx to give Ix′ , the x component of I convolved with the derivative of the Gaussian, and convolve Iy with Gy to give Iy′ , the y component of I convolved with the derivative of the Gaussian.
5. Compute the magnitude of the edge response by combining the x and y components. The magnitude of the result can be computed at each pixel (x, y) as:

M(x, y) = √ { Ix′ (x, y)2 + Iy′ (x, y)2 }

1. Implement a non-maximum suppression algorithm. Pixels that are not local maxima should be removed with this method. In other words, not all the pixels indicating strong magnitude are edges in fact. We need to remove false-positive edge locations from the image
2. Apply Hysteresis thresholding to obtain final edge-map

The implementation has the following methods implemented for the process mentioned above:

1. Reading an image and converting it to a matrix using OpenCV
2. Creating a gaussian mask with filter size 3 and standard deviation as provided by the user
3. Obtaining a derivative of the gaussian filter created in step 2
4. Convoluting the image with the gaussian filter along X and Y axes– made use of padded images by mirroring the edge pixels
5. Convoluting the above results with the derivative of the gaussian filter
6. Computing the magnitude and direction of the image using the formula mentioned above for magnitude. For direction, a method in NumPy is used as follows:

θ(p) = arctan2(Iy′ , Ix′ )

1. After obtaining the magnitude and direction, the Non-Maximum Suppression (NMS) algorithm has been implemented to discard the false positive edges. The NMS sharpens the edges which otherwise are blurred out due to smoothening. For e.g. –

A picture containing black

Description automatically generated 

After smoothening After NMS

|  |
| --- |
| 1. After applying NMS to the image, performed the hysteresis thresholding to create the edge map. Provided a low and high threshold, any pixel below the low threshold is discarded from an edge and any pixel above the high threshold is part of an edge 2. For the pixels ranging between the low and high threshold values, we explore whether the given pixel is a neighbor of any pixel that already belongs to an edge and mark it as an edge pixel accordingly 3. To perform this exploration, the implementation of a depth-first search algorithm is used. Pixels connected to the edge pixels are colored White to indicate them as edge 4. After performing the Hysteresis thresholding, we get the result of our edge detector algorithm. For e.g.   Diagram, engineering drawing  Description automatically generated with medium confidence |
| **Observations:**  **The standard deviation –** The experiments were done with 5 sigma values viz. 1.0, 3.0, 5.0, 7.0 and 9.0. The edge detector detects even the micro edges with a lower sigma value, but as we go higher, sharp edges are not detected. Examples can be seen below:  A picture containing text, stone  Description automatically generated A picture containing text, building  Description automatically generated A picture containing text, building, tower  Description automatically generated  3.0 5.0 7.0  **A picture containing outdoor object  Description automatically generated A picture containing outdoor object  Description automatically generated**  3.0 5.0  **Results:**  Below is an example of the result after each step performed during the edge detection process:  A picture containing weapon, light, dark  Description automatically generated A picture containing ground, plane, outdoor, floor  Description automatically generated **A picture containing outdoor object, web  Description automatically generated** **A picture containing black, web  Description automatically generated** Ix Iy dIx dIy  A picture containing black  Description automatically generated Engineering drawing  Description automatically generated Diagram, engineering drawing  Description automatically generated with medium confidence Magnitude NMS Hysteresis  Parameters for the above result:  1. Standard Deviation: 5.0 2. Threshold:  a. Lower: 25  b. Higher: 55 |
|  |