

**ADLD Course Project**

**Team no: -17**

|  |  |
| --- | --- |
| **Name** | **USN** |
| Daneshwari Hurali | 01FE19BEC270 |
| Girija Rathod | 01FE20BEC409 |
| Chaitra Hiremath | 01FE20BEC426 |

**Problem Statement: -**

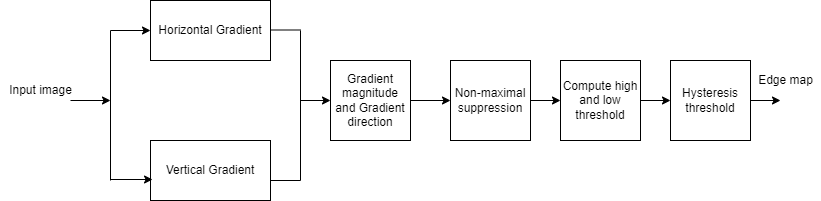
“Canny Edge Detector On FPGA”

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986.

* Edges are the sudden and significant changes in the intensity of an image. These changes happen between the boundaries of object in an image.
* Why edge detection?

To understand the shape of the objects in the image.

**Architecture: -**



**Algorithm: -**

**Step1: -** Conversion of the input image to Grayscale.

**Step2: -** Gaussian Blur operator is used to remove noise in the filter.

**Step3: -** Intensity gradient calculation-

Sobel filter is used (it has 2 kernels one is x(horizontal) and other is y(vertical)).

These kernels are convoluted with original image through which edge points are calculated.

Idea behind Gaussian filter is the center having more weight than the rest.

1-D gaussian filter is convoluted with x-derivative to find Sobel-x (Gx)and -D gaussian filter is convoluted with y-derivative to find Sobel-y. (Gy).

The next sequence is to convolute the Gx and Gy over the input image which enables to calculate the value for one pixel at a time. Then the column and row-shift happen. The above calculations shall help the reader in visualizing the edges.

G= sqrt(Gx^2 + Gy^2)

The edge direction is =inv tan (Gy/Gx)

**Step4: -** Non-maximum suppression-

Enables us to derive thin edges from thicker edges. The subsequent step is to relate the identified edge direction to a direction that can be sketched in the image I.e., ideally it is a prediction of how the movement of edges could happen.

The non-maximum suppression as the name suggests a process where suppression of the pixels to zero which cannot be considered as an edge is carried out. This enables system to generate a thin line in the output image.

**Step5: -** Thresholding -

There is still some noise. Some edges could be missed in the to address this challenge thresholding is followed.

Where we take high threshold value and low threshold value. Any pixel value above high threshold value is seen as strong edge and any pixel value below low threshold value is not an edge at all hence set them to 0.

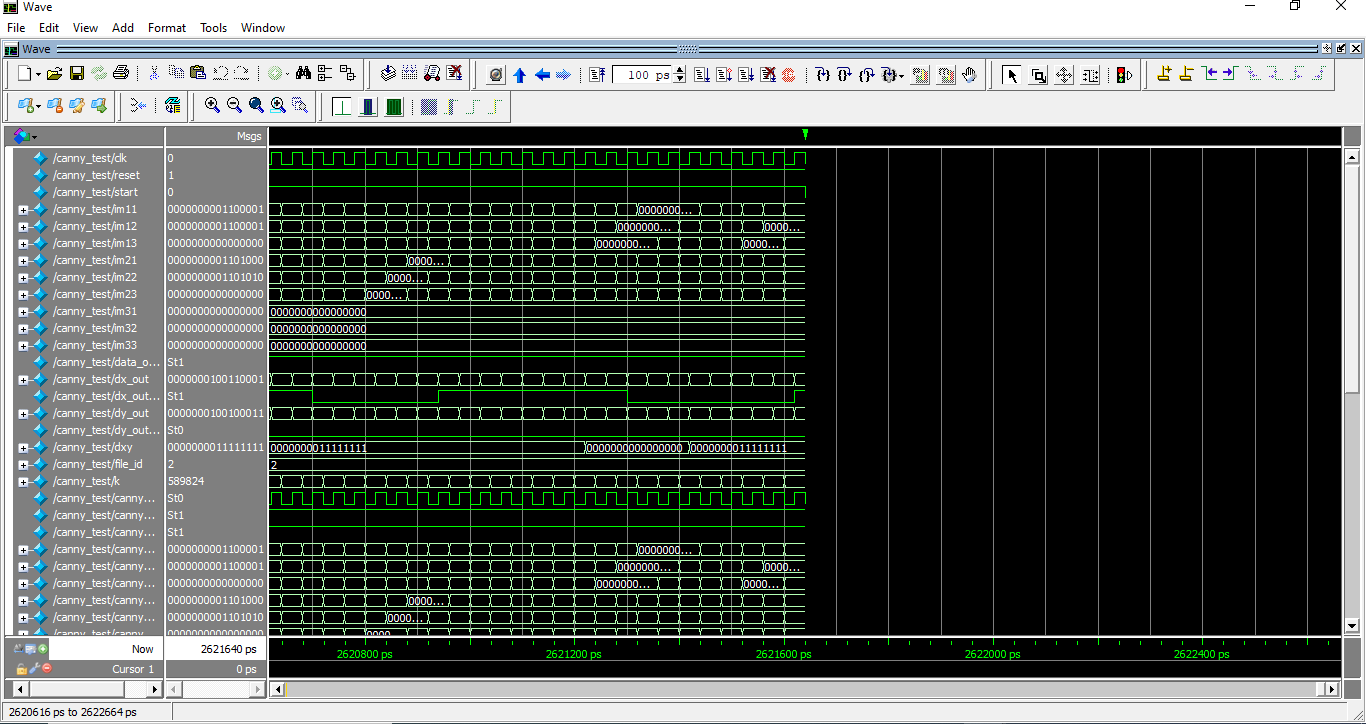
The values in between low and high threshold values may or may not be edges. They are referred as weak edges.

**Step6: -** Edge Tracking

To know which of the weaker edges are actual edges. Weak edges connected to strong edges are called as strong/actual edges and are retained. Weak edges which are not connected to stronger ones are to be removed**.**

**Step7: -** The final cleansing-

All the remaining weak edges are removed once the process is complete, we get the edged output.

**Simulation: -** Waveform

**Output: -**



Input image



Gray image



Canny Edged Output