Setting up the Raspberry Pi board and implementing the Canny edge detection Algorithm

(Orion Peter Fernandes – 65047854)

1. Project Description

Goals and Objective

The main goal of the project is to demonstrate working of the Canny Edge detector on hardware implemented by a Raspberry Pi 4. This project involves several objectives including –

- Setting up of the Raspbian OS image on the Raspberry Pi
- Setting up Wi-Fi and Webcam Devices for the Raspberry Pi
- Enabling and capturing live images with the camera through the Raspberry Pi
- Compiling the provided Canny C code to create the Canny executable
- Performing trials on the PGM image with applied variations in variables.
- Reviewing impact of each parameter with respect to image.

  The Canny Edge detection executable shall perform edge detection on the PGM image as per input parameters.

2. Experimental Setup

Methodology

- Raspbian OS was pre-installed on SD card during installation as provided during ECPS 206.
- System was configured for WI-Fi and viewing through VNC Viewer as part of ECPS 206.
- Raspberry Pi was accessed through VNC Viewer using Mobile Hotspot connectivity.

- ‘ canny_local.c ’ and ‘ test.pgm ‘ files downloaded from Canvas Assignment 1 page
- Files transferred to Raspberry Pi via USB from PC
- New directory created for ECPS 204 HW1 and files copied to it
- Downloaded canny code is compiled with some warnings after changing to source directory which creates an executable

  gcc canny_local.c -lm -o ./canny

- Test PGM image is executed to provide output image with command –

  ./canny test.pgm 1.0 0.2 0.6

- Image of hills downloaded from the internet and named “hills”
- Downloaded image converted to PGM format online through website - https://convertio.co/\
- Converted image transported to Raspberry Pi via USB and executed upon.

3. Results

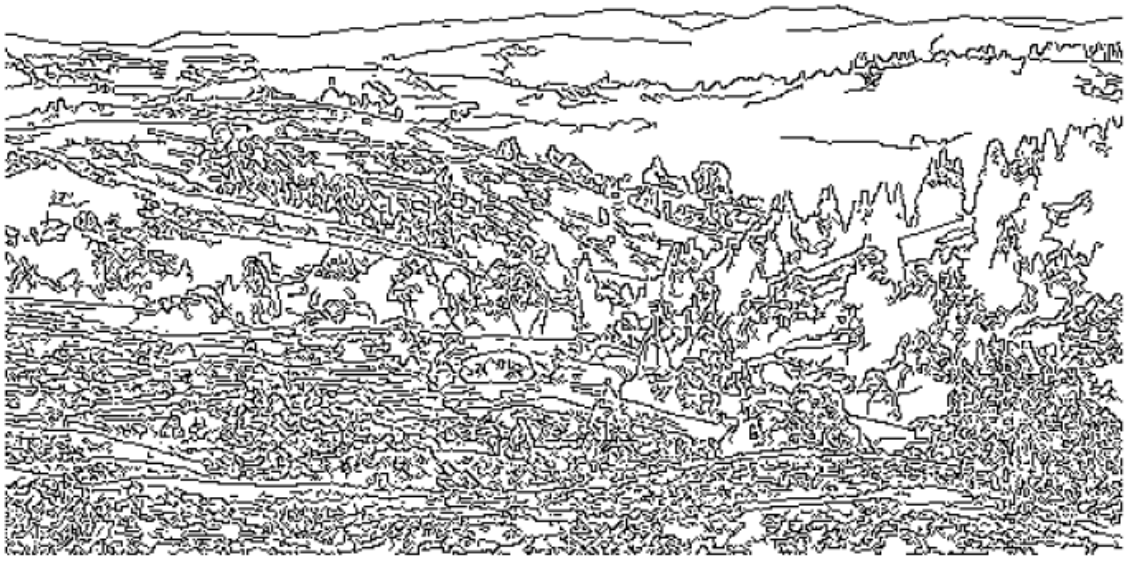
- Raspberry Pi was set up successfully with Wi-Fi compatibility and enabled for use through VNC Viewer
- Canny edge detection was successfully performed on the given test PGM image and an image from the internet.
- Execution performed with different parameters changed with each trial and results noted.

hills.jpg – Original Image

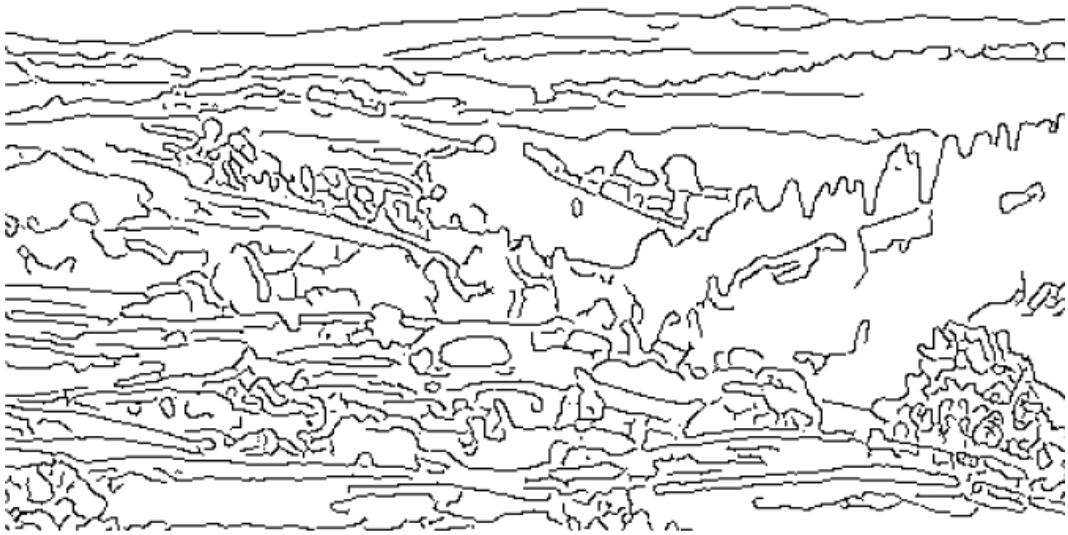
hills.pgm – PGM format

Trial 1 – Variations in Sigma

Sigma – 1.0 tlow – 0.2 thigh – 0.6



Sigma – 0.6 tlow – 0.2 thigh – 0.6

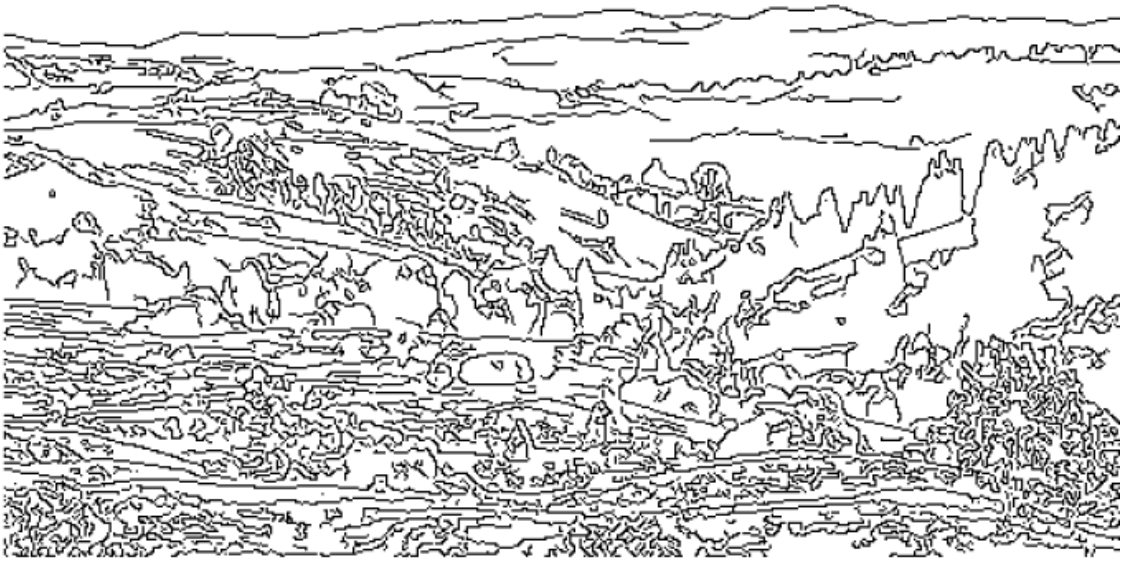


Sigma – 1.8 tlow – 0.2 thigh – 0.6

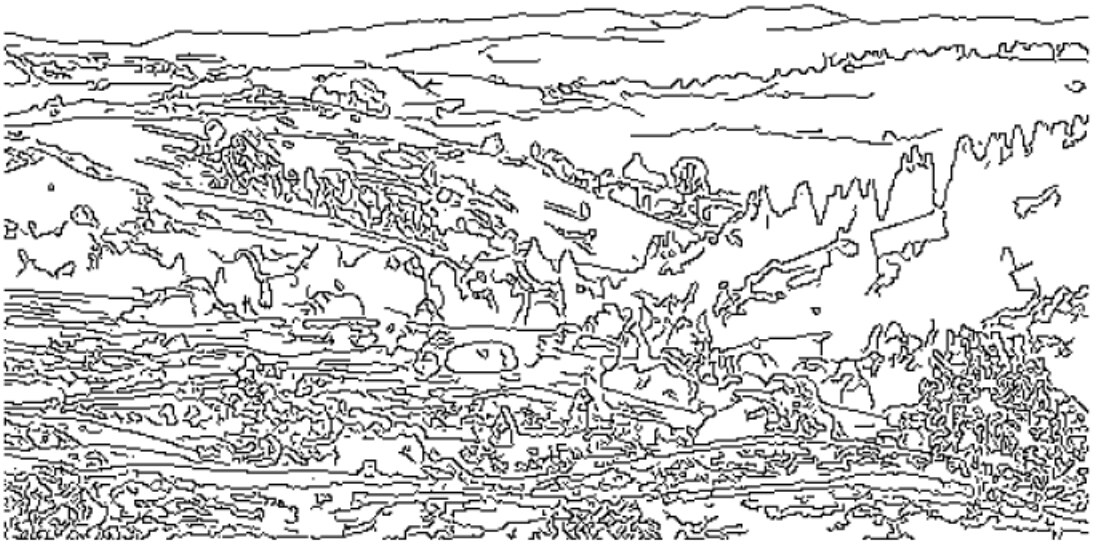
Sigma – 2.4 tlow – 0.2 thigh – 0.6

Trial 2 - Variations in tlow

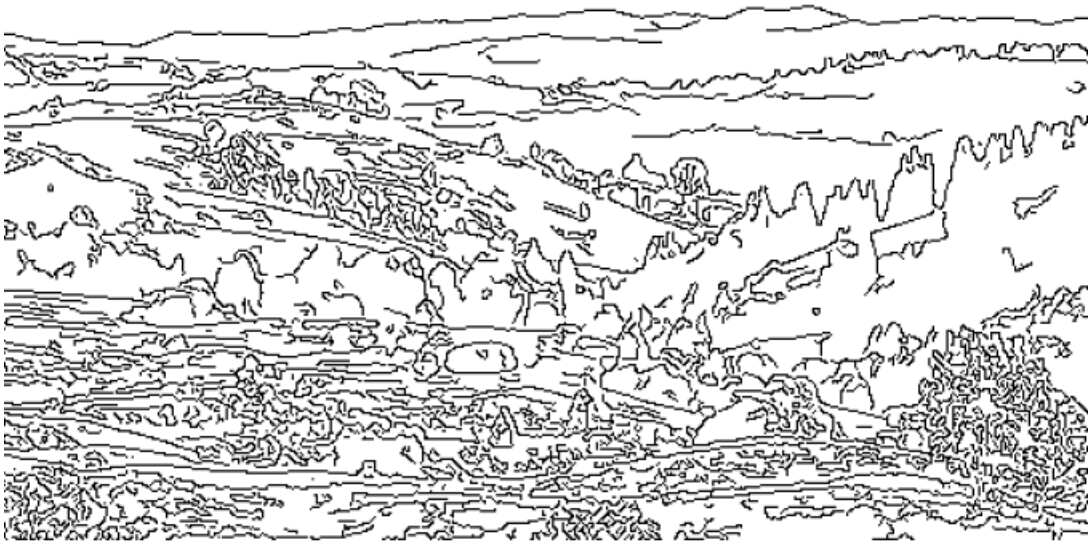
Sigma – 1.0 tlow – 0.2 thigh – 0.6



Sigma – 1.0 tlow – 0.3 thigh – 0.6



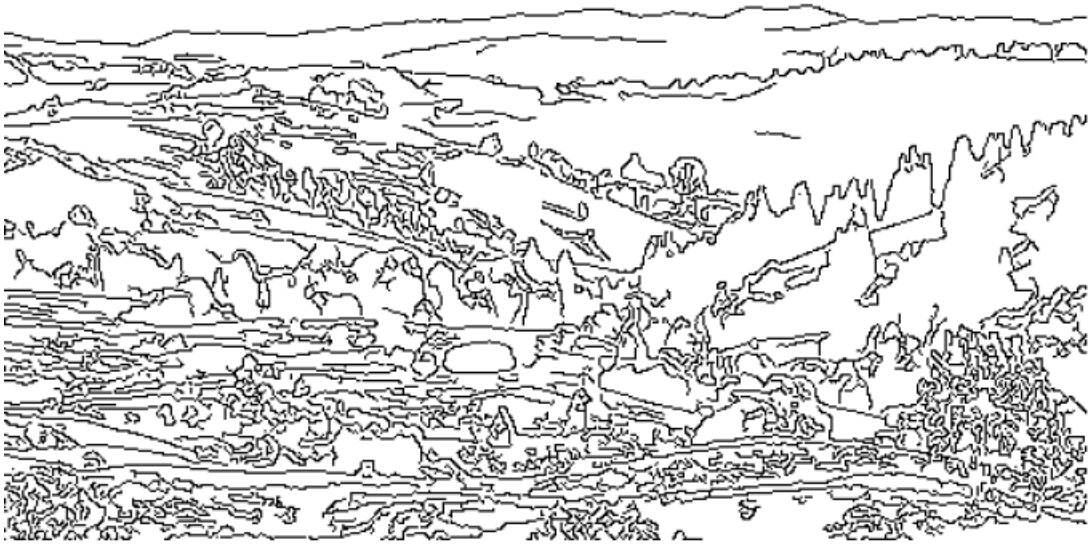
Sigma – 1.0 tlow – 0.4 thigh – 0.6



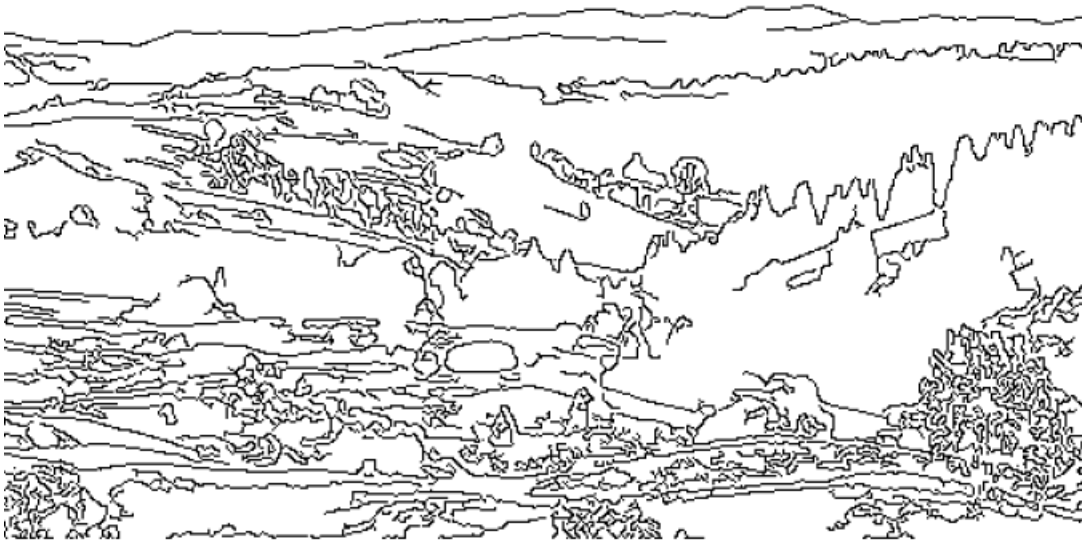
Sigma – 1.0 tlow – 0.5 thigh – 0.6

Trial 3 – Variations in thigh

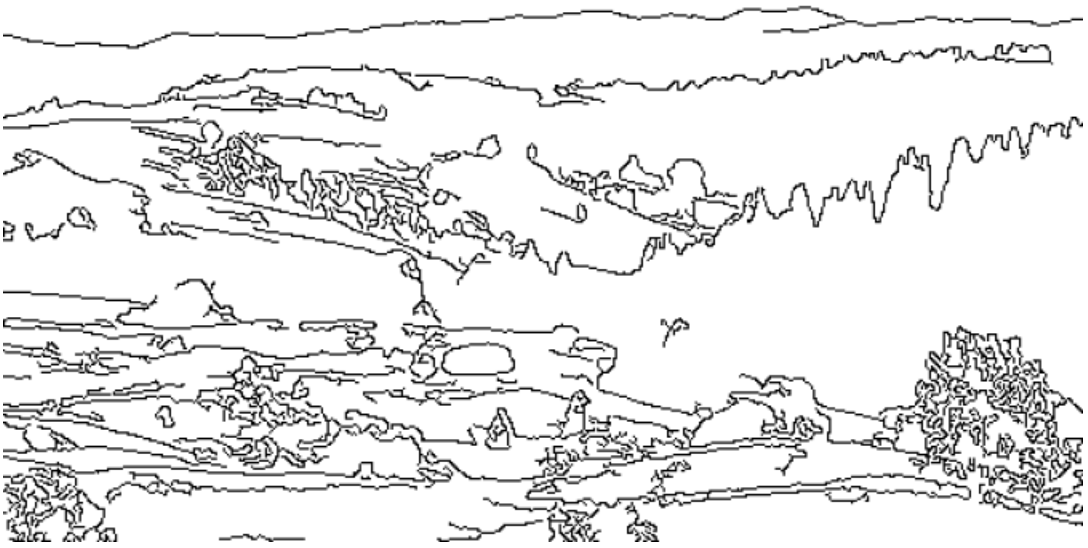
Sigma – 1.0 tlow – 0.2 thigh – 0.6



Sigma – 1.0 tlow – 0.5 thigh – 0.7



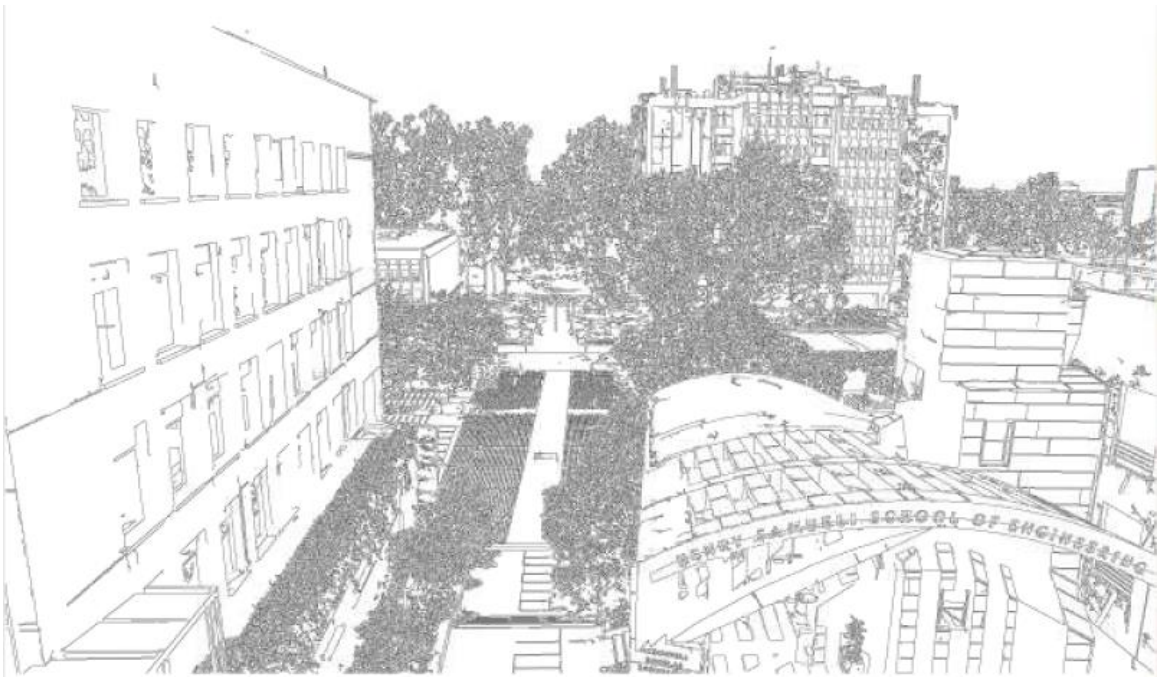
Sigma – 1.0 tlow – 0.5 thigh – 0.8



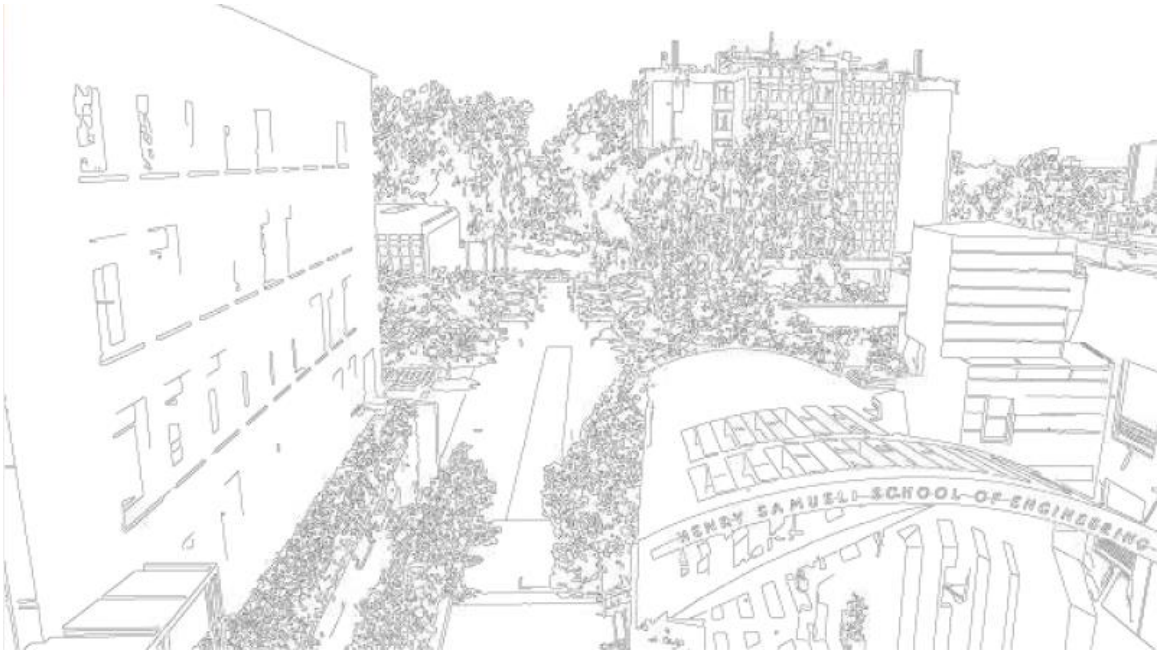
Sigma – 1.0 tlow – 0.5 thigh – 0.9

Trial 4 – Performed on Test.pgm

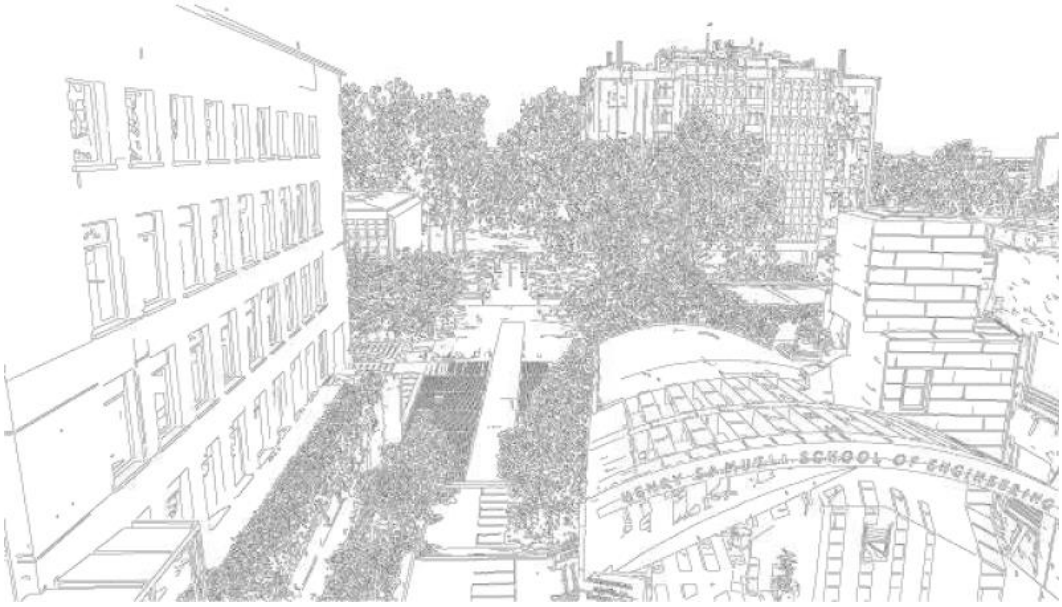
Test.pgm – Original



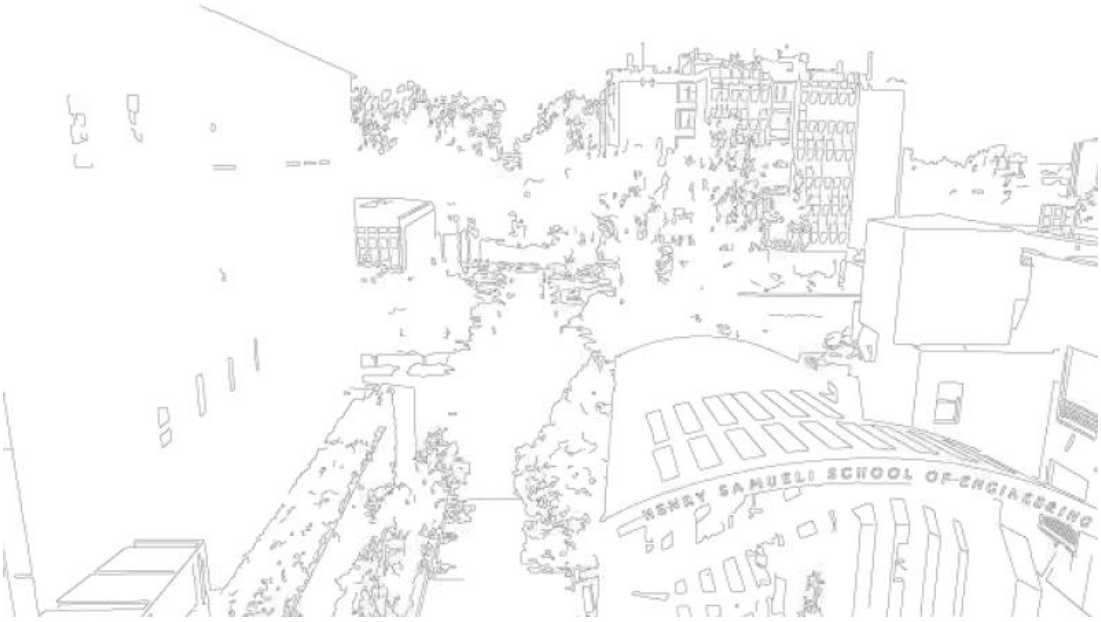
Sigma – 0.7 tlow – 0.1 thigh – 0.7



Sigma – 2.0 tlow – 0.3 thigh – 0.8



Sigma – 1.6 tlow – 0.5 thigh – 0.6



Sigma – 2.2 tlow – 0.4 thigh – 0.9

4. Problems and Discussion

Challenges and Solutions

- Problem with transporting downloaded files from PC to Raspberry Pi
  Solution – Using a USB drive

- Issue with obtaining PGM format for example image
  Solution – Image converted on website - https://convertio.co/\

- From the results procured, they seem to match my first expectations of the Canny Edge detector.

- The results agree with my first expectations since the Canny Edge Detector processed the input PGM image to produce an image with edges devoid of any color or shadows thus fulfilling its application as an edge detector.

- Methodology can be improved by using pre-defined set of parameters for Edge Detection of an image or set of images and setting up a WebDAV server for transporting images to the Raspberry Pi

- Website used for PGM image conversion - https://convertio.co/ . It supports more than 300+ formats, is fast and easy to use and provides custom settings for each conversion.

Discussion

- The value of sigma is determined by standard deviation of the Gaussian smoothing filter. Decreasing the value causes more edges to be present in the output image while reducing the value decreases definition losing some edge.

- The value of tlow is the high threshold edge strength value whose low value to be used in hysteresis (fraction: 0-1). Increasing the value causes a slight reduction in edge detection.

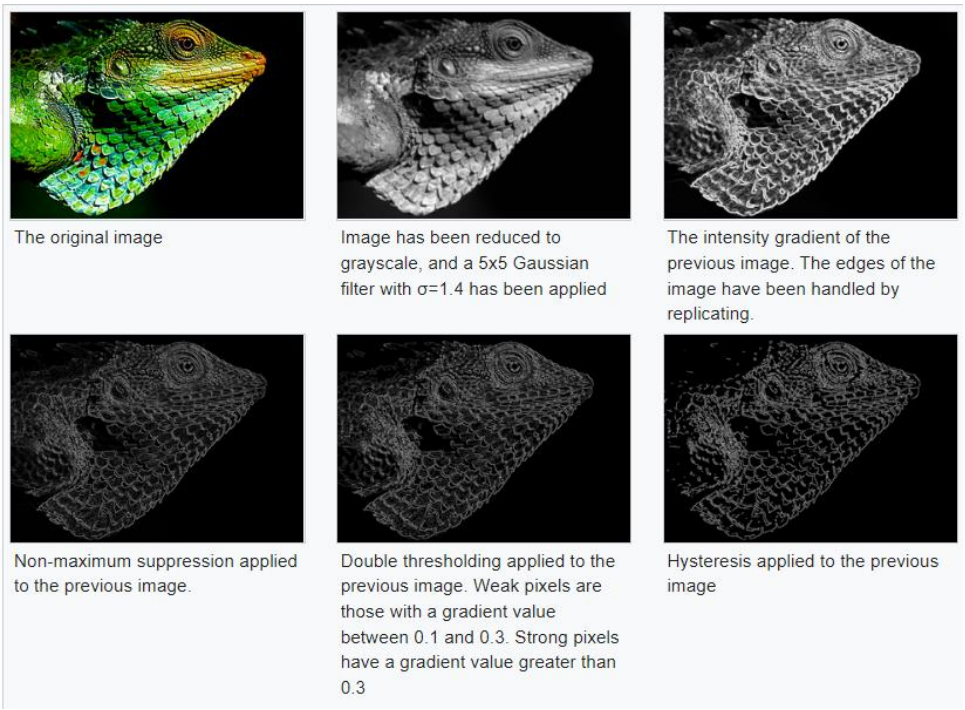
- The value of thigh is the percentage of the gradient of the magnitudes in a histogram whose high value is to be used in hysteresis. Increasing the value causes deletion of edges that are closely spaced to each other.

5. Conclusion

The project “Setting up the Raspberry Pi board and implementing the Canny edge detection algorithm” performed as expected and with varied results. The edge detection operator uses a multi-stage algorithm to detect a wide range of edges in images. Accuracy depends on the number of edges that can be detected and shown in the output image.

The process of Canny Edge detection involves different steps –

1. Apply Gaussian Filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge to hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.



Source: https://en.wikipedia.org/wiki/Canny_edge_detector