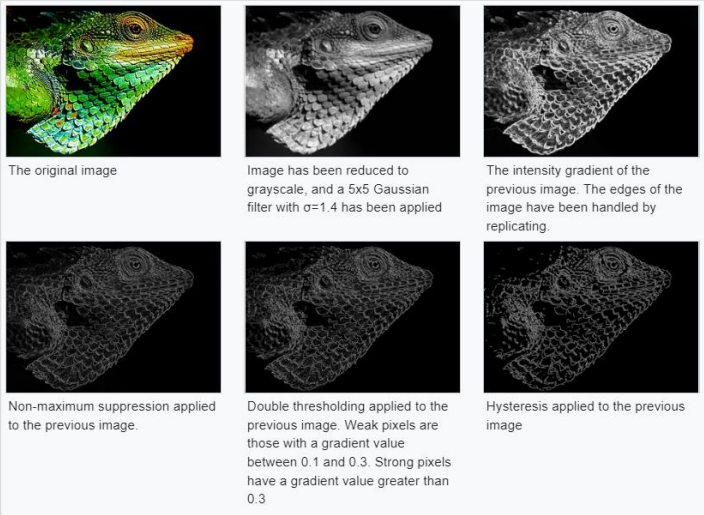
Image Edge Detection

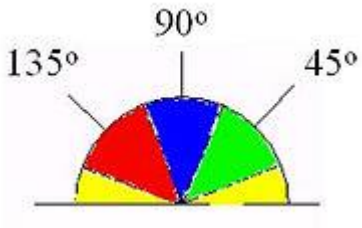
Image Edge Detection is a fundamental image analysis process that allows the extraction of the outline of objects within a 2D image. It has varied uses across a wide spectrum from machine vision and machine learning algorithms to biomedical imaging, making it an interesting choice for an integrated hardware parallel processing project.



The Canny image edge detection is an edge detection operation that uses a multistep algorithm to process the image and extract the edges. Each step of the image detection algorithm provides an opportunity for digital acceleration and possible parallelisation.

The process first applies a Gaussian filter over the whole image to remove potential noise. This could possibly be done as a local Gaussian filter to decrease any loss of resolution.

The second process is finding the intensity gradient of the image and is done in 4 separate directions. This could allow digital acceleration through parallel processing each of the 4 directions independently.



The third process is to use suppression or thresholding to any spurious gradients created.

We can then apply a double threshold to the image to identify valid edges and remove any other weaker edge elements that may not be connected.

Project specifications

- USB Bus simulated to transfer image files to and from the hardware accelerator and computer.
- Files will be converted from jpg to txt files using MATLAB with each item in the text file corresponding to a pixel value.
- The text file will be read by the Verilog program and preform edge detection using the Canny Edge Detection Algorithm.

- The Verilog program will then send the output edge detected file as a text file to MATLAB that will then convert the file back to a jpg file.
- Golden measure written in Python using existing libraries to compare speed with hardware acceleration.

Project Goals

- Send Files between simulated FPGA and computer.
- Upload image to computer via GUI.
- Image edge detection run successfully.
- Image of edges is returned to the user via the GUI.

Acceptance Criteria

- Suitable visual discernment of image edges
- Acceptable accuracy of detected edges
- Low noise of final output image
- Time of processing improvements over the golden measure

References

Columbia University. (n.d.). Retrieved from https://www.youtube.com/watch?v=hUC1uoigH6s

Fischer, R. (n.d.). *University of Edinburgh*. Retrieved from https://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm

OpenCV. (n.d.). Retrieved from https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

Scikit. (n.d.). Retrieved from https://scikit-image.org/docs/stable/auto_examples/edges/plot_canny.html

Wikipedia. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Canny_edge_detector