March 18, 2015

A01952442

Adam Tew

# Edge Detection Report

### Introduction:

Edge detection is a fundamental component in image processing. In general terms an edge in an image is defined as "a significant local change in the image intensity" (MachineVision). Thus edge detection is a mathematical algorithm used to identify those changes of intensity. This article discusses edge detection algorithm analysis.

### Background:

Image detection is not respective to any single language so we have the luxury of analyzing algorithms and ideas instead of specific implementation with a given language. This article focuses on some specific modules for Python that work well with image manipulation - namely OpenCV, NumPy, and SciPy. There were many more, but these appear to be the most capable of performing edge detection.

Even with capable software to read, write, and manipulate images, there is still need to overcome the obstacle of mathematical evaluation. "Edge detection is the name for a set of mathematical methods which aim at identifying points in a [digital image](http://en.wikipedia.org/wiki/Digital_image) at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*."

Edge detection is not always a simple matter because of the math. As stated by M. Sarifuddin, "Detecting edges in digital images is a tricky operation in image processing since images may contain areas with different degrees of noise, blurring and sharpness. Such operation represents an important step of the whole process of similarity shape analysis and retrieval." Thus we need algorithms to enact correction for certain images.

### Experimental setup/Algorithms:

Some of the known algorithms I found were the Canny Edge Detection, Laplactian and LoG (Laplacian of Gaussian), Robert, Prewitt, and Sobel algorithms. Under multiple studies (Avlash, Monica), it is stated that the Canny algorithm is the best performing of the rest, however it also comes at the cost of being the most computationally expensive. That being said, here is more detail on all of them.

The "Canny Edge Detection is a popular edge detection algorithm. It was developed by John F. Canny in 1986." (Mordvintsev, Alexander)The Canny Edge algorithm is considered the best, and attached is an example using OpenCV from their website. The algorithm includes the following steps:

1. Noise Reduction
2. Finding Intensity Gradient of the Image
3. Non-maximum Suppression
4. Hysteresis Thresholding

For a brief description of the others:

The Laplacian filters use derivatives. Since derivatives are very susceptible to noise it's normal to first apply a smoothing filter, like the Gaussian filter, before using the Laplacian. When used consecutively they make up the Laplacian of Gaussian (LoG) operation.

The Prewitt operator, named after Judith M. S. Prewitt, is a discrete differentiation operator used with the intensity of gradients and is considered a relatively inexpensive algorithm. The operator is "based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical directions." (Prewitt, J).

The Sobel operator, named after Irwin Sobel, is also a discrete differentiation operator that performs computations of gradients. The operator detects edges "based on applying a horizontal and vertical filter in sequence. Both filters are applied to the image and summed to form the final result. The two filters are basic convolution filters of the form." (RoboReam).

The Roberts cross operator, named after Lawrence Roberts, was one of the first edge detectors, initially proposed in 1963. This algorithm approximates the gradient of an image by "computing the sum of squares of the differences between diagonally adjacent pixels." (Davis, LS).

### Results and conclusions:

There are many modules and libraries used for edge detection and image correction in Python. Also, any of the aforementioned algorithms can be functionally implemented using those Python libraries.

In conclusion, Canny's edge detection algorithm is the one to choose for performance analysis. "It has been shown that the Canny’s edge detection algorithm performs better than all these operators under almost all scenarios." (Maini, Raman).

### References:

<http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html>

<http://en.wikipedia.org/wiki/Edge_detection>

<http://scipy-lectures.github.io/advanced/image_processing/>

<http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter5.pdf>

<http://www.ncbi.nlm.nih.gov/pubmed/18255489>

<http://www.owlnet.rice.edu/~elec539/Projects97/morphjrks/laplacian.html>

<http://link.springer.com/chapter/10.1007%2F978-0-387-72500-0_12#page-2>

<http://docs.opencv.org/modules/imgproc/doc/filtering.html>

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.927&rep=rep1&type=pdf>

<http://academic.mu.edu/phys/matthysd/web226/Lab02.htm>

<http://en.wikipedia.org/wiki/Prewitt_operator>

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>

<http://en.wikipedia.org/wiki/Sobel_operator>

<http://en.wikipedia.org/wiki/Roberts_cross>

<http://www.academia.edu/5380026/PERFORMANCES_ANALYSIS_OF_DIFFERENT_EDGE_DETECTION_METHODS_ON_ROAD_IMAGES>