Reverse Engineering Graphs: Obtaining Data Points from Scatter Plots

Maya Wei

Computer Systems Lab, 2008 – 2009

October 31, 2008

Abstract

Various programs exist to take data points and use them to render a graph. However, once the data are put into visual form, there is a loss of numerical information if the original data cannot be obtained. This project seeks to take data from a graph; in essence, the purpose is to reverse engineer a given graph. This will provide for a set of data points which can be used for various other numerical purposes, not simply the graph form in which they are presented.

Introduction

A scatterplot is a visual representation of bivariate data. However, once the data are in this visual form, there is minimal further testing one can do. If only given a graph, there is no way to perform statistical tests on dots laid along an image. By being able to reverse-engineer a graph – to be able to look at a graph and be able to calculate the what point the coordinate represents – the computer's capabilities could simulate the human mind with more efficiency.

The purpose of this experiment is to create a fully working program that will be able to correctly identify the points on a graph. Given a graph in png format, the program will calculate the point's relation to the axes and give its theoretical coordinate.

If data points are obtained, it would facilitate various statistical calculations that need hard data. It could serve as a double-checking device statistical calculations which do not show work.

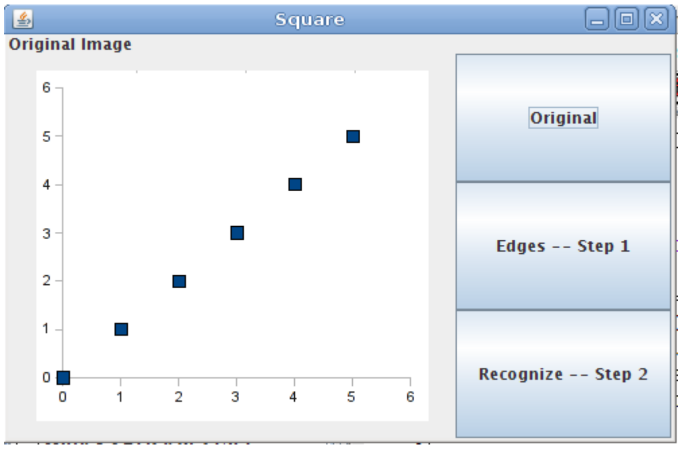
Background

This program utilizes various simplistic forms of image analysis. The principle of image analysis is based in edge detection – finding points which do not blend in with the background image. There are minimal exact replications of reading scatterplots; current day computer vision programs take many factors into consideration such as slant, skew, color, depth, etc. However, this graph is a 2D surface, so much of the complexity of current day research is minimally applicable.

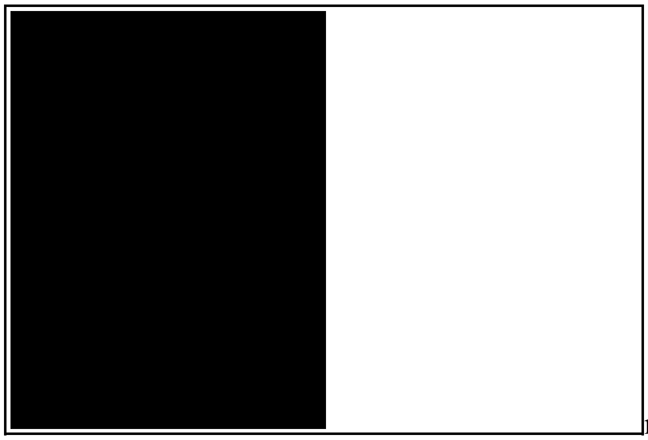
In a recent study, Pordesimo, Columbus, Batchelor, and Methuku developed a plug-in for ImageJ which would allow them to correctly identify shapes and size according to aspect ratio, rectangularity, and major axis ratio. When they applied their identification strategies to images of food grains and ground Miscanthus particles, the mean deviations were less that 1.3%.

Development Sections

The project is divided into four parts: display, edge detection, distinguishing edges, and identification. The display is based off of a png image of a graph created in OpenOffice Calc. It's points are at (0,0), (1,1), (2,2), (3,3), (4,4), and (5,5). There is no background and no gridlines – the graph has as little extraneous information as possible.

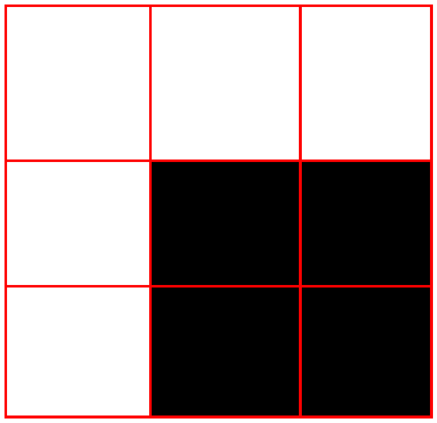


Displayed is the original image. The GUI was created recently in order to be able to toggle between original image and edge detection images more easily.
In the display section, there were also tests to guarantee that the BufferedImage could correctly manipulate the image. The following image was used:

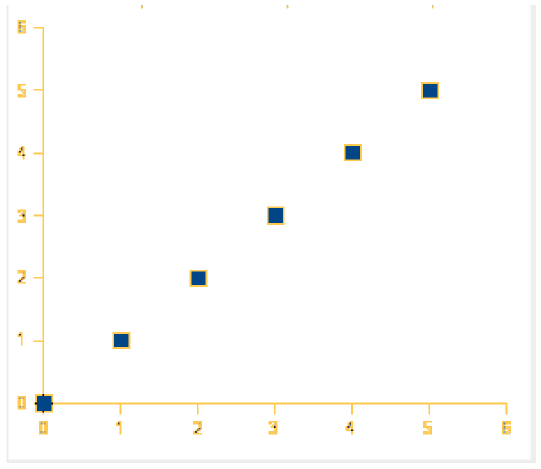


The BufferedImage correctly identified the points at which the image changed from black to white. PixelGrabber, another method in Java, also did, but was more unwieldy.

The second step to identify edges. Currently, the edge detection method in this program is highly primitive. It takes the color of the point at (0, 0) and makes the assumption that the point's color is also the background color. It then looks at each individual point (x, y) and the points' neighbors (x+1, y), (x – 1, y), (x, y + 1), (x, y – 1). If any of the points' neighbors are the same color as (0, 0), then that means that (x, y) is touching an edge. For example, the black squares at (1, 1), (2, 1), and (1, 2) would be considered edges. The bottom rightimage shows the graph with edges highlighted.



The modified image appears below.



Shape recognition has yet to be implemented. However, with the edges known, it will now be possible to group the points into axis, number, and point categories. This might be possible with identifying segment lengths – if there is one single long line, then it is most likely an axis. Identifying the points from the Once the center of each "point" is found, it will be possible to determine the "point"s relation to the axis.

Testing is very visual right now; if there is an error, the image will most likely be highlighted incorrectly. Incorrect coloring will be especially prevalent when distinguishing points between axes between numbers is attempted.

Results

As of the current, there is minimal to conclude. The expected outcome is a program that will effectively identify points. In second and third quarter, it is expected that the program will continue and further fulfill its purported goal.

Bibliography

Faure and Vincent. Document Image Analysis For Active Reading. ACM International Conference Proceeding Series; Vol. 259. p. 7 – 14. 2007.

Igathinathine, Pordesimo, Columbus, Batchelor, and Methuku. Shape identification and particles size distribution from basic shape parameters using ImageJ. *Computer and Electronics and Agriculture*, p. 168-182. Fall, 2008.