Reverse Engineering Graphs: Obtaining Data Points from Scatter Plots
Maya Wei
Computer Systems Lab 2008 – 2009

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 while being an exploration of various methods of visual analysis.

Background: The project deals in basic methods of image analysis and shape differentiation. The process used relies much on edge detection, which can be from very basic techniques to very advanced algorithms. In addition, factors such as connectivity and image recognition are used. The graph used was created in OpenOffice Calc; there are no gridlines and no colored background. The graph itself is very minimalist; the graph used for this exploration are points located at (1, 1), (2, 2), (3, 3), (4, 4), and (5, 5) (see Figure 1). The image is saved as png and read with Java’s ImageReader.

Edge Detection – the first process involves the filtering of points into a single large ArrayList<int[]>. A simple method of edge detection is used. The algorithm takes the color of the pixel at (0, 0), and assuming it to be the background color for the entire graph, the rest of the points are compared to (0, 0). Edges are defined under this criteria:

1) not the same color as (0, 0)
2) touching at least one point of the color (0, 0) in a horizontal, vertical, or diagonal fashion

When these requirements are found, the points are grouped from there into ArrayLists of points that are connected to each other; the container is ArrayList<ArrayList<int[]>> groups.

Interpretation – Three ArrayList<ArrayList<int[]>> are created – points, axis, and numbers. This step attempts to divide the whole array of edge points to their respective titles: points, axis line / tick marks, and numbers. Using the longest length ArrayList within groups as the axis (which is an assumption), all those point groups with members to the left of or below the axis are considered in numbers, where the others (to the right or above) are considered points. The graphical point (0, 0), located on the axis, is something of difficulty. Using the center of each ArrayList within points, the relation to the axes can be determined (see Figure 2).

From there, using image analysis and comparison, the scale in numbers can be identified (see Figure 3).

Further Development: This program has several steps that could be expounded upon and clarified, such as a better method for edge detection. A more efficient method of number identification can also be developed.