Austin New
Arabnia Homework 1 Report

My experimentation on the Digital Differential Analyzer and Bresenham's Algorythm for drawing lines in a digital environment showed decisively the superiority of the Bresenham Algorythm. Though, my results were not as 'night and day' as I was led to expect. My test used the timer provided by the SDL (Simple Direct Media) package. For my experimentation I iterated 100,000 random lines 275 times. The statistical results are below:

100,000 Lines Drawn	Average Time(ms)	Lines per ms	Variance	Standard Deviation
Bresenham	1376.1309090909	72.6675052056	876.2455739881	29.6014454713
DDA	1589.8905109489	62.8974129422	921.4971257453	30.3561711312

Bresenham's algorithm plots 10 more lines per millisecond than the Simple DDA and does so with marginally less variation on time. This lower variance and standard deviation may be caused from the height to width ratio provided, 1280x800.

Some of the difference in my findings between the Bresenham's algorithm and Simple DDA algorithm may caused by my particular implementation of the algorithms. My decision to cast the floating point error value in the DDA algorithm to an int rather than explicitly truncating it, and to represent both the DDA and Bresenham's algorithms in single functions (rather than assigning a certain iteration function at the initialization, based on the orientation of the line, removing some conditional statement and memory overhead). In this representation Bresenham's algorithm is superior to the DDA in only its disuse of floating point representation. The DDA uses, almost always, less additions and shifts, less conditional statement overhead, and less memory access and modification. Even with all these disadvantages Bresenham's representation of the problem in a purely integer space is clearly better computationally.

Bresenham	DDA
1352	1610
1378	1595
1355	1605
1368	1617
1380	1575
1378	1574
1349	1571
1382	1582
1416	1555
1377	1568
1401	1563
...	...

A sample of my data.