

A Run-Length Slice Line Drawing Algorithm without Division Operations

Khun Yee Fung, Tina M. Nicholl, and A. K. Dewdney

Department of Computer Science
Middlesex College
University of Western Ontario
London, Ontario, Canada N6G 5B7

Abstract

Of the two major approaches to line drawing, run-length slice algorithms are seldom used because of the division operation deemed necessary in these algorithms. The biggest advantage of these algorithms, the reduction of additions used, is considered outweighed by the division used. In this paper, a new run-length slice algorithm that does not require a division operation is presented. Furthermore, it uses the double-stepping paradigm in incremental line drawing algorithms to reduce the number of additions used by at least half.

For sufficiently long lines, this algorithm uses at least 50% fewer arithmetic operations than Wu et al.'s bi-directional double-step incremental algorithm. But because of its high initialization cost, for short lines, it is less efficient. For a line with endpoints $(0,0)$ and $(\delta x, \delta y)$, the strategy is then to use the bi-directional Bresenham algorithm for very short lines ($\delta x < 20$), the bi-directional double-step algorithm for moderate long lines ($20 \leq \delta x \leq 110$), and the new algorithm for the longer lines ($\delta x > 110$).

Keywords: Algorithm, raster graphics, scan conversion, line drawing.

1. Introduction

In computer graphics, there are two major approaches to drawing line segments (henceforth simply called lines) on a raster device using only integer operations. One is the *incremental method*, drawing a fixed number (usually one) of pixels, in order, from one end of the line to the other. The selection of each successive pixel is based on the value of a variable. The decision is subsequently used to update the value of the variable. The selection of the next pixel is then based on the value of the updated variable and so on. The other approach is to compute the number of pixels with the same ordinates or abscissas, the runs, to be drawn at the same time. These are the *run-length slice* line drawing algorithms [2][6].

The original integer incremental line drawing algorithm is the Bresenham line drawing algorithm [1]. Wu et al [10] use double incremental steps to speed up the Bresenham algorithm by a factor of two by considering all possible patterns for three consecutive pixels. Using symmetry, lines can be drawn from both ends at once to speed up by another factor of two [4][7].

The major reason for not using run-length slice algorithms is the division operation deemed necessary for these algorithms. This operation is considered to be too expensive, as it outweighs the reduction in the number of additions.

In this paper, a new run-length slice algorithm is presented. This algorithm also uses the double step and bi-directional paradigms to lower the number of arithmetic operations. The division deemed to be necessary