AN APPLICATION OF EUCLID'S ALGORITHM TO DRAWING STRAIGHT LINES

by C.M.A.Castle & M.L.V.Pitteway

ABSTRACT

An algorithm is proposed which uses Euclid's Algorithm to control two production rules which can construct the "best-fit" incremental line. The advantages of its implementation are discussed.

Introduction

The increasing use of non-vector graphics output peripherals, (such as raster scan devices and incremental plotters) has accelerated the search for efficient algorithms to produce "best-fit" straight lines. Effectively, on a raster scan device, each straight line must be "approximated" by a quantised sequence of diagonal and/or square movements, and the "best-fit" is obtained by applying the constraint that after each incremental step, the pixel nearest to the "real" line is the one selected for illumination.

Using this criterion, it is possible to select the appropriate pixel by using either a floating-point differential analyser (Newman & Sproull), or a form of Bresenham's integer based algorithm (Stockton 1963, Bresenham 1965). Sproull has shown that the Bresenham algorithm can be derived from the differential analyser, thus establishing that both generate identical output strings.

A significant property of both of these algorithms is that, in general, they both require one test per output move.

The Output of the Algorithm

Even a cursory glance at the output of a "best-fit" algorithm reveals a profound symmetry which is at first sight most surprising.

If,for example, a line is drawn from (0,0) to any pair of prime co-ordinates (u,v) with u>v>0, the output is a palindrome, symmetrical about the central element.

eg from (0,0) to (23,7) produces:

S D SS D SSS D SS D SS D SSS D SS D S

(Spaces added to improve readability)

NATO ASI Series, Vol. F17
Fundamental Algorithms for Computer Graphics
Edited by R. A. Earnshaw

CONTROLLING THE PRODUCTION RULES WITH EUCLID'S ALGORITHM

Analysis of the output of a "best-fit" algorithm suggests that it is a regular language. Furthermore, it is a language which when u and v are prime, is of the form:

$$L = L^{-1} \quad \text{(ie a palindrome)}$$

This strongly suggests that any move concatenation technique controlled by Euclid's algorithm , would require production rules of the form:

$$R := T.R^{-1}$$

(Where A.B represents string A contatenated with string B)

Fig (1) shows the algorithm which employs Euclid's algorithm as a discriminator between two symmetric production rules. If a "best-fit" line is to be drawn from (0,0) to (u,v) (with u>v>0 to keep within the first octant), then the initial selection for a and b would be:

b := v (the number of diagonal moves to be made)
a := u-v (the number of square moves)

Move 1 has an initial value of S, move 2 has an initial value D

Eg The following table represents the output of the algorithm in constructing the line (0,0) to (51,11)

a	b	Move 1	Move 2
40	11	S	SD
29	11	S	SDS
18	11	S	SSDS
7	11	SSDSS	SSDS
7	4	SSDSS	SSDSSSDSS
3	4	SSDSSSDSSSSDSS	SSDSSSDSS
3	1	SSDSSSDSSSSDSS	SSDSSSDSSSSDSSSSDSSSDSS
2	1	SSDSSSDSSSSDSS	SSDSSSDSSSSDSSSSDSSSDSSSSDSSSSDSSSDSS
1	1		

OUTPUT move2(move1)$^{-1}$ repeated a times, giving:

SS D SSS D SSSS D SSSS D SSS D SSSS D SSSS D SSS D SSSS D SSSS D SSS D SS

In this example, the complete 51 move "best-fit" output stream was determined in just 8 tests. If the selected values of u and v share a common factor, then the algorithm will generate the "best-fit" output up to the point where the "real" line passes through a pixel. This pattern is then to be repeated the highest common factor number of times.

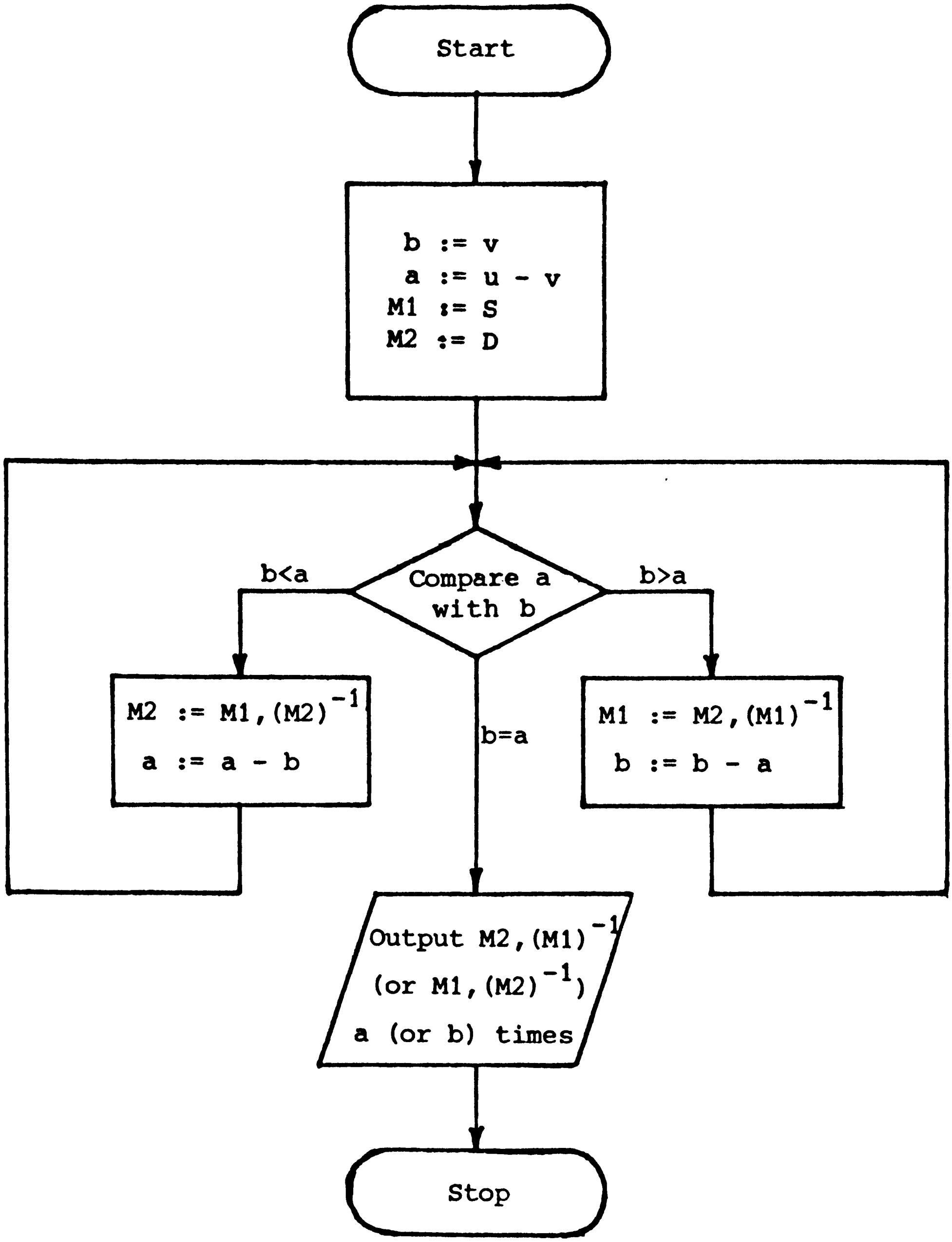

Fig (1) The algorithm for generating "best-fit" output, with Euclid's algorithm acting as a discriminator between two production rules. ($u > v > 0$, the output in other octants being produced by symmetry).

(A comma denotes the concatenation of two ouput strings, and $(Mx)^{-1}$ denotes the reversal of the string Mx.)

This type of pattern is produced because:

(1) The "real" line only passes through "best-fit" pixels at its two ends (and nowhere else).
(2) Appeal to symmetry requires that the output associated with drawing from either end to the centre must be indistinguishable.

If u and v are not themselves prime, but can be expressed as a prime multiple of a common factor, it is only necessary to construct the "best-fit" output up to the point where the "real" line passes through a pixel; thereafter the same pattern will be regenerated, and the regenerated pattern will itself show reflected symmetry. More generally, following the removal of any common factor, if the remaining numbers are not themselves prime, reflection may be about a double element.

The line from (0,0) to (28,6), for example, produces:

SS D SSS D SSSS D SSSS D SSS D SSSS D SS

This can be re-written as:

```
SS D SSS (DS) SSS D SS   |  SS D SSS (DS) SSS D SS
                         |
          |              |            |

          A              B            C
```

B is the place where the "real" line passes through pixel (14,3)
A & C are the double element about which the sub-strings are symmetrical.

The importance of any common factor to the form of the output has been noted by Earnshaw (1978). In 1982, Pitteway & Green presented an algorithm which employed Euclid's algorithm to trap the common factors. These were then used to drive appropriate move concatenation techniques which improved the operational speed of Bresenham's Algorithm, while still retaining its basic stucture.

In this paper , we show that Euclid's algorithm ALONE can be used to control the appropriate output sequence for an integer based "best-fit" generator. This effects a further saving in arithmetic from the algorithm of Pitteway & Green (1982).

Implementation

Any hardware which is capable of Left or Right rotation of one register into another, can perform the move transposition operation in just one machine cycle. (fig 2) This would help to make implementation of this algorithm very efficient.

```
Regiister 1      S S S S D S S S S D S S S

Register 2              S S S D S S S S D S S S S
```

Fig (2). Register 1 contains the current state of movel. Register 2 contains the result of performing $(\text{movel})^{-1}$

References

(1) EUCLID, Elements (Book 7) Proc. Platonic Acad. c.350BC Vol 4009

(2) STOCKTON, F.G. (1963) Algorithm 162 CACM, Vol 6, No. 4

(3) BRESENHAM, J.E. (1965) Algorithm for computer control of a digital plotter. IBM Systems Journal, Vol 4, No 1, pp 25-30

(4) PITTEWAY, M.L.V. & GREEN, A.Bresenham's Algorithm with run line coding shortcut. The Computer Journal, Vol 25, No 10

(5) EARNSHAW, R.A. Line tracking with incremental plotters. The Computer Journal Vol 23, No 1

(6) NEWMAN, W.M. & SPROULL,R.F. Principles of interactive computer graphics 1981 McGraw Hill pp 23-28