# Incremental Line Compaction

**Jack E. Bresenham**
IBM System Products Division, IBM UK Laboratories Ltd, Mailpoint 183, Hursley House, Hursley Park, Winchester, Hampshire SO21 2JN, UK

**Raster devices, such as digital plotters, CRT or plasma panel displays, and matrix or ink jet printers, represent 'straight' lines in quantized fashion as a sequence of unit axial and unit diagonal steps. Dichotomous run lengths and periodic repetitive patterns in these incremental or digital lines provide a basis by which the step sequences for quantized lines can be treated in compressed form for storage or transmission. Earnshaw recently published a paper describing an investigation of two compaction alternatives encoding either run lengths or repeated patterns. This paper describes a simple algorithm to incorporate both run length and repeated pattern encoding for step sequence compaction. Also illustrated is the similarity in form of the repetitive loop used to generate either runs or single steps and either full lines or periodic patterns; initial parameter values differ, but the subsequent iterative process is identical.**



## INTRODUCTION

Raster devices, such as digital plotters, CRT or plasma panel displays and matrix or ink jet printers, represent 'straight' lines in quantized fashion as a sequence of unit axial and diagonal steps. Dichotomous run lengths and periodic repetitive patterns in these incremental or digital lines provide a basis by which the step sequences for quantized lines can be treated in compressed form for storage or transmission.

Earnshaw recently published papers describing an investigation of two compaction alternatives encoding either run lengths or repeated patterns.[1,2] This note describes a simple algorithm to incorporate both run length and repeated pattern encoding for step sequence compaction without the necessity to calculate separately the explicit greatest common factor of the delta $X$ and delta $Y$ displacements of the line from $(X_1, Y_1)$ to $(X_2, Y_2)$.

For full line encoding, a partial first octant modification ($0 < \tan\theta < 0.5$) of Bresenham's original algorithm[3,4] for single step sequence generation and a comparable Freeman/Reggiori-like algorithm[5,6,7] for run length generation are used. The termination test in each of the two full line algorithms is then changed to generate only the first full period of the step or run sequences. Substructure within the fundamental period as described by Cederberg[6] and Brons[8] is not treated here.

To the best of the author's knowledge, Freeman first observed that evenly spaced run length slices should be present in incremental lines.[9] Reggiori[7], working with Freeman at NYU, first published a run length algorithm and described a very good encoded run length compaction scheme. Alternative derivations of incremental line run length properties include Refs 1, 6, 8, 10, 11 and 12.

## RATIONALE

Incremental line single step or run sequences, in uncompacted or compacted form, can be generated with essentially the same short iterative loop which uses only addition and sign testing. Subsequent expansion and reconstruction of the encoded line representations can be done by devices having only counters and equality testing for arithmetic and logic capability. Enlargement by an integer scale factor $E$ readily can be incorporated.

To illustrate the methods, consider the incremental form of the first octant line from $\mathbf{S} = (0, 0)$ to $\mathbf{T} = (45, 6)$. Let $\mathbf{a}$ represent one single step in the axial unit direction, $\mathbf{a} = (1, 0)$ in the first octant, and $\mathbf{d}$ represent one single step in the diagonal unit direction, $\mathbf{d} = (1, 1)$ in the first octant. The step sequence for the line from $\mathbf{S}$ to $\mathbf{T}$ is

| | | | | | | |
|---|---|---|---|---|---|---|
| **aaad** | **aaaaaaad** | **aaaaaad** | **aaaaaaad** | **aaaaaad** | | |
| 3 | 7 | 6 | 7 | 6 | | |
| | | | | | **aaaaaaad** | **aaa** |
| | | | | | 7 | 3 |

and the step sequence for its complementary image from $\mathbf{S}' = (0, 0)$ to $\mathbf{T}' = (45, 39)$ is

| | | | | | | |
|---|---|---|---|---|---|---|
| **ddda** | **ddddddda** | **dddddda** | **ddddddda** | **dddddda** | | |
| 3 | 7 | 6 | 7 | 6 | | |
| | | | | | **ddddddda** | **ddd** |
| | | | | | 7 | 3 |

This suggests a 'step encoding—full line sequence' compaction form in which the two unit direction moves are referenced as $\mathbf{s}_1$ and $\mathbf{s}_2$ with a binary valued step sequence $\{H_i\}$ of length $\Delta A$ in which 1 represents an $\mathbf{s}_1$ move and 0 represents an $\mathbf{s}_2$ move. The full line from $\mathbf{S}$ to $\mathbf{T}$ can be encoded

$$\mathbf{s}_1 = 1, 0 \qquad \mathbf{s}_2 = 1, 1 \qquad \Delta A = 45$$

$$\{H_i\} = 1110 \quad 11111110 \quad 1111110 \quad 11111110 \quad 1111110 \quad 11111110 \quad 111$$

and its complementary image from $\mathbf{S}'$ to $\mathbf{T}'$ will differ only in that $\mathbf{s}_1 = 1, 1$ and $\mathbf{s}_2 = 1, 0$.

For a 'run encoding—full line sequence' compaction form, one again can reference unit direction moves as $\mathbf{s}_1$ and $\mathbf{s}_2$, but define an encoded run length sequence $h_s$, $\{H_i\}$, $h_t$ of length $\nabla B + 2$ for explicit $\mathbf{s}_1$ run length designation with one single $\mathbf{s}_2$ step implied between each explicit $\mathbf{s}_1$ run. Sequence interpretation will be as follows.

CCC–0010–4620/82/0025–0116 $02.50

If $h_s$ or $h_t = 0$ one has an $s_1$ run of length $M = \lfloor Q \div 2 \rfloor$

$= 1$ one has an $s_1$ run of length $M - 1$

If $\{H_i\} = 0$ one has an $s_1$ run of length $Q = \lfloor \Delta A \div \nabla B \rfloor$

$= 1$ one has an $s_1$ run of length $Q - 1$

where

$$\Delta A = \max[|\Delta X|, |\Delta Y|]$$

and

$$\nabla B = \min[|\Delta X|, |\Delta Y|, |(|\Delta X| - |\Delta Y|)|]$$

The full line from S to T can be encoded

$s_1 = 1, 0$  $s_2 = 1, 1$  $Q = 7$  $\nabla B = 6$
$h_s = 0$  $\{H_i\} = 010101$  $h_t = 0$

and its encoded complementary image from S′ to T′ will differ only in that $s_1 = 1, 1$ and $s_2 = 1, 0$.

Since delta $X$ and delta $Y$ displacements in the example lines have a common factor greater than one, further compaction can be obtained by encoding only the basic period of the line [i.e. the line from the origin to either (45, 6) or (45, 39) amounts to three repetitions of the step sequence one would use from the origin to (15, 2) or (15, 13)]. The periodic codings from S to T would be

'step encoding—periodic pattern sequence'

$s_1 = 1, 0$  $s_2 = 1, 1$  $\Delta A = 45$  $I = \Delta a = 15$
$\{H_i\} = 1110\ \ 11111110\ \ 111$

'run encoding—periodic pattern sequence'

$s_1 = 1, 0$  $s_2 = 1, 1$  $\nabla B = 6$  $I = \nabla b = 2$  $Q = 7$
$h_s = 0$  $\{H_i\} = 01$  $h_t = 0$

The encoded complementary image from S′ to T′ again will differ only in that $s_1 = 1, 1$ and $s_2 = 1, 0$.

Figure 1 outlines the four encoding schemes:

step encoding—full line sequence
step encoding—periodic pattern sequence
run encoding—full line sequence
run encoding—periodic pattern sequence

with reference to appropriate transformation Tables 1–6 and the iterative stepping loop of Figs 2 and 3. Incremental line step reconstruction from the encoded forms is shown in Figs 4 and 5, and includes provision to apply an integer enlargement scaling factor $E$. Reconstruction of unidirectional or null lines is obvious and hence not explicitly flowcharted.

## COMMENTS

The iterative stepping loop flowcharts include treatment of the equal error instance ($\nabla = 0$ of which there will be $g$ occurrences when $\Delta A = g\Delta a$ and $\nabla B = g\nabla b$ with $\Delta a$ and $\nabla b$ relatively prime and $\Delta a$ is even and $\nabla b$ is odd) which provides an exactly reversible path. The method differs from that of Boothroyd and Hamilton but produces a comparable effect.[13]

As shown in Table 6, the greatest common denominator (GCD) of delta $X$ and delta $Y$ (or $\Delta A$ and $\nabla B$) easily can be obtained as a by-product from the periodic pattern stepping loop of Fig. 3 by counting the number of times, $I$, the loop is traversed. Of course, the GCD could be employed in lieu of $\Delta A$ or $\nabla B$ in the 'step-periodic' or 'run-periodic' compaction formats. Using either Earnshaw's method or the techniques described here, one must calculate a line's incremental step sequence. The former approach employs separate calculation of the GCD, while in the latter the GCD can be obtained from the incremental step sequence calculations which are common to either approach.

If $g$ is the greatest common denominator of $\Delta A$ and $\nabla B$ and either form of run encoding is being used, one should note that the final element in the sequence $\{H_i\}$ is used only through the first $(Eg - 1)$ periodic repetitions or $(E - 1)$ full line repetitions. The test $\nabla B = K$ in the flowchart in Fig. 5 causes the iteration to exit from evaluation of $\{H_i\}$ before the last element when the whole sequence $\{H_i\}$ is not to be repeated again, e.g. 0{010101}0 represents (0, 0) to (21, 15) in a 'run length encoding—full line sequence'. The final 1 in {01 01 01} is not decoded in Fig. 5. {One also can note that the $\{H_i\}$ encodings for the lines from (0, 0) to (45, 6) and from (0, 0) to (21, 15) are identical since their first partial octant loop parameters match even though their run lengths, $Q$, differ.} The encoding $\{H_i\}$ takes this form since one can see that the above can be further compacted to 0{01}0 representing (0, 0) to (21, 15) in the 'run length encoding—periodic pattern sequence'. The re-construction re-cycles round the {01} by re-zeroizing $J$ each time. But again, one only uses {01 01 01} and the final (expanded) 1 once again remains undecoded. In this way the required number of reconstructed increments are obtained. The last element

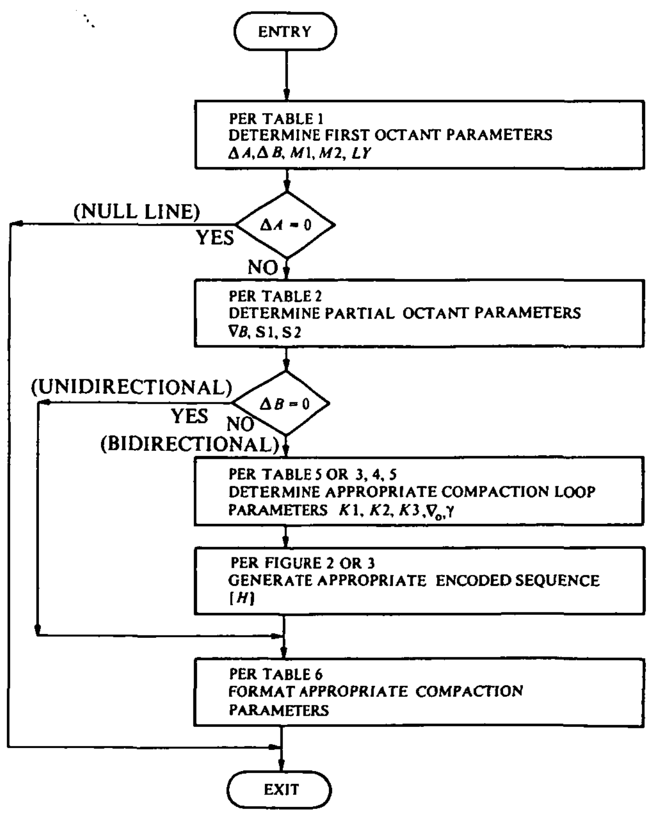


**Figure 1.** Incremental line compaction.

**Table 1. First octant normalization**

| $\Delta X$ | $m_{21}$ | $\Delta Y$ | $m_{22}$ | $LY$ | $\Delta Z$ | $\Delta A$ | $\Delta B$ | $m_{11}$ | $m_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| $\geq 0$ | 1 | $\geq 0$ | 1 | 1 | $\geq 0$ | $\lvert\Delta X\rvert$ | $\lvert\Delta Y\rvert$ | $m_{21}$ | 0 |
| $<0$ | $-1$ | $<0$ | $-1$ | 0 | $<0$ | $\lvert\Delta Y\rvert$ | $\lvert\Delta X\rvert$ | 0 | $m_{22}$ |

where line is from integer starting coordinate point $(X_1, Y_1)$ to integer terminating coordinate point $(X_2, Y_2)$ and
$\Delta X = (X_2 - X_1)$ $\Delta Y = (Y_2 - Y_1)$ $\Delta Z = (\lvert\Delta X\rvert - \lvert\Delta Y\rvert)$

**Table 2. Partial first octant normalization**

| $\Delta B$ | $\nabla B$ | $s_1$ | $s_2$ |
|---|---|---|---|
| $\leq \lvert\Delta Z\rvert$ | $\Delta B$ | $(m_{11}, m_{12})$ | $(m_{21}, m_{22})$ |
| $> \lvert\Delta Z\rvert$ | $\lvert\Delta Z\rvert$ | $(m_{21}, m_{22})$ | $(m_{11}, m_{12})$ |

**Table 3. Run length parameters**

| | |
|---|---|
| $Q = \left\lfloor \frac{\Delta A}{\nabla B} \right\rfloor$ | floor of $\Delta A \div \nabla B$<br>quotient of integer divide |
| $R = {}_{\nabla B}\vert^{\nabla A}$ | $\Delta A$ modulo $\nabla B$<br>remainder of integer divide for $Q$ |
| $M = \left\lfloor \frac{Q}{2} \right\rfloor$ | floor of $Q \div 2$<br>truncated, single right shift of $Q$ |
| $N = {}_{\lvert 2\nabla B}\vert^{\Delta A}$ | $\Delta A$ modulo $2\nabla B$<br>$N = R$ if $Q$ even (no underflow in right shift for $M$)<br>$N = R + \nabla B$ if $Q$ odd (underflow in right shift for $M$) |

**Table 4. Initial/final run length encodings**

| $N$ | $\Delta Y$ | $h_s$ | $h_t$ |
|---|---|---|---|
| $\neq 0$ | $\geq 0$ | 0 | 0 |
| $\neq 0$ | $<0$ | 0 | 0 |
| $=0$ | $\geq 0$ | 0 | 1 |
| $=0$ | $<0$ | 1 | 0 |

**Table 5. Compaction loop initialization**

| Compaction form | K1 | K2 | $V_0$ | $\gamma$ | K3 Full line | K3 Periodic pattern |
|---|---|---|---|---|---|---|
| Single steps | $2\nabla B$ | $2(\nabla B - \Delta A)$ | $2\nabla B - \Delta A$ | $LY$ | $\Delta A$ | $2\nabla B - \Delta A$ |
| Run lengths | $2R$ | $2(R - \nabla B)$ | $N + 2(R - \nabla B)$ | $\sim LY$ | $\nabla B$ | $N + 2(R - \nabla B)$ |

**Table 6. Requisite compaction parameters**

| | $\Delta A$ | $\nabla B$ | $s_1$ | $s_2$ | $I$ | $Q$ | $h_s, h_t$ | $\{H_i\}$ | Number of elements in $\{H_i\}$ |
|---|---|---|---|---|---|---|---|---|---|
| Unidirectional | × | | × | | | | | | |
| Steps—full line | × | | × | × | | | | × | $\Delta A$ |
| Steps—periodic pattern | × | | × | × | × | | | × | $I = \Delta a$ |
| Runs—full line | | × | × | × | | × | × | × | $\nabla B$ |
| Runs—periodic pattern | | × | × | × | × | × | × | × | $I = \nabla b$ |

Note: greatest common denominator of $\Delta A$, $\Delta B$, $\nabla B$ is available as
steps—periodic pattern gcd $= \Delta A \div I = g$
runs—periodic pattern gcd $= \nabla B \div I = g$

in $\{H_i\}$ is, of course, redundant in that its associated length could be calculated as the sum of the initial and terminating lengths coded as $h_s$ and $h_t$. It is convenient to carry the final element for enlargement or repeated periodic pattern cycling simplicity.

Table 6 indicates parameters which must be stored in addition to the line's binary encoding. For very short lines, these parameters would cause a stored representation to be longer than simply storing the single steps directly. For longer lines and for lines in which delta $X$ and delta $Y$ have a common factor greater than 1, the stored representation can effect significant savings. In a specific application, one would want to observe the 'cost' of executing reconstruction as a distributed module and to consider such effects as housekeeping for byte or word boundary breakage in the binary encoding. The trade-offs of space vs execution time, parametric coding vs direct step usage, and partitioned or distributed function execution are not considered here.

## Acknowledgments

The contribution of Ms Sharon Hamilton in preparing artwork and Ms Dorothea Lowendick and Ms Janet Lennard in preparing the manuscript is gratefully acknowledged. Helpful as well were stimulating discussions with Ms Nancy Bull and Dr Rae Earnshaw, and comments on improving clarity from the referees.

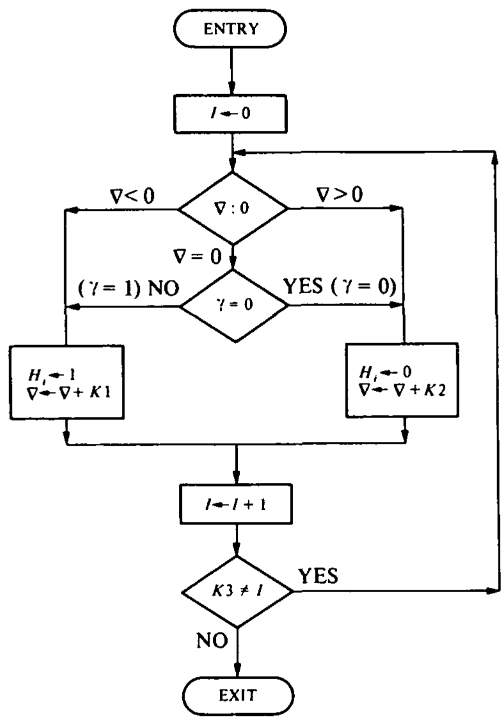


**Figure 2.** Full line sequence.

| | K1 | K2 | K3 | $\nabla_0$ | $\gamma$ |
|---|---|---|---|---|---|
| For single step compaction: | $2\nabla B$ | $K1 - 2\Delta A$ | $\Delta A$ | $K2 + \Delta A$ | $LY$ |
| For run length compaction: | $2R$ | $K1 - 2\nabla B$ | $\nabla B$ | $K2 + N$ | $\sim LY$ |

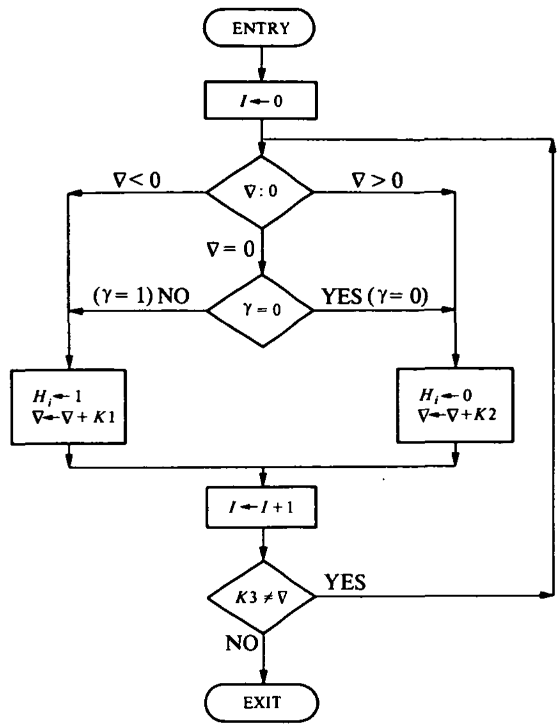


**Figure 3.** Periodic pattern sequence.

| | K1 | K2 | K3 | $\nabla_0$ | $\gamma$ |
|---|---|---|---|---|---|
| For single step compaction: | $2\nabla B$ | $K1 - 2\Delta A$ | $K2 + \Delta A$ | K3 | $LY$ |
| For run length compaction: | $2R$ | $K1 - 2\nabla B$ | $K2 + N$ | K3 | $\sim LY$ |

BIDIRECTIONAL - SINGLE STEPS: RECONSTRUCTION

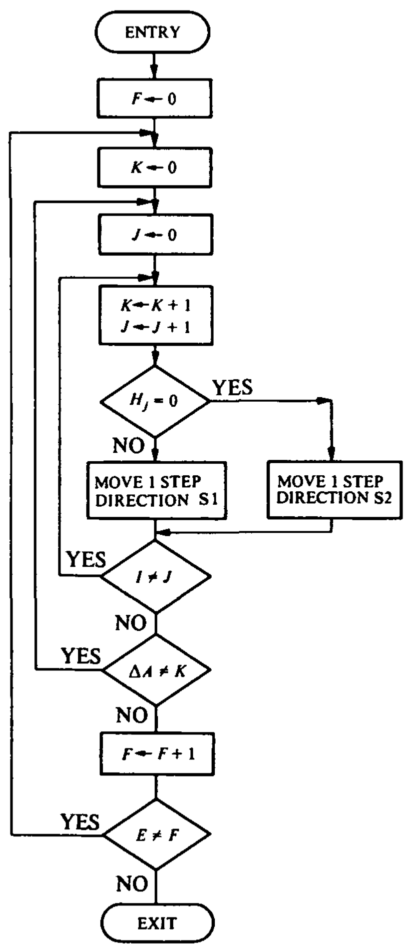


**Figure 4.**

For full line: Set $I \leftarrow \Delta A$

For full line or periodic pattern: Allow integer enlargement scaling $E$

$$E \in [1, 2, 3, \ldots]$$

BIDIRECTIONAL - RUN LENGTHS: RECONSTRUCTION

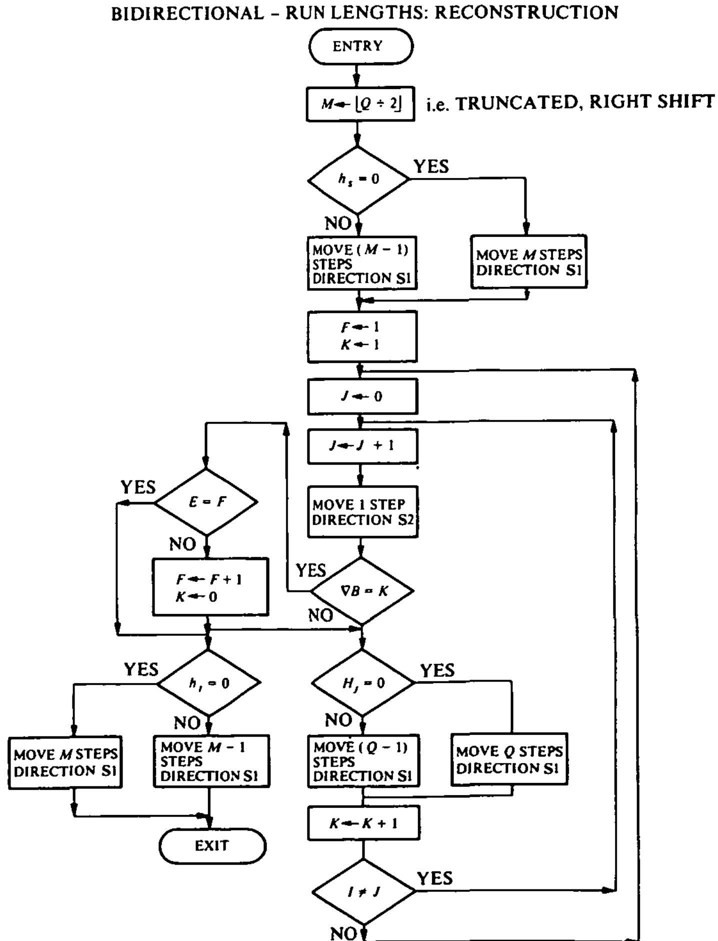


**Figure 5.**

For full line: Set $I \leftarrow \nabla B$

For full line or periodic pattern: Allow integer enlargement scaling $E$

$$E \in [1, 2, 3, \ldots]$$

## REFERENCES


1. R. A. Earnshaw, Line tracking for incremental plotters. *The Computer Journal* **23** (No. 1), 46–52 (February 1980).
2. R. A. Earnshaw, Line generation for incremental and raster devices. *Computer Graphics* **11** (No. 2), 199–205 (Summer 1977—SIGGRAPH '77 Proceedings).
3. J. E. Bresenham, An incremental algorithm for digital plotting. *ACM National Conference* (August 1963).
4. J. E. Bresenham, Algorithm for computer control of a digital plotter. *IBM Systems Journal* **4** (No. 1), 25–30 (January 1965).
5. J. E. Bresenham, D. G. Grice and S. C. Pi, Run length slices for incremental lines. *IBM Technical Disclosure Bulletin* **22–8B**, 3744–3747 (January 1980).
6. Roger L. T. Cederberg, A new method for vector generation: *Computer Graphics and Image Processing* **9** (No. 2), 183–195 (February 1979).
7. G. B. Reggiori, *Digital computer transformations for irreguar line drawings*, pp. 46–61. Technical Report 403–22, New York University (April 1972). Available from US Department of Commerce as AD–745–015.
8. R. Brons, Linguistic methods for the description of a straight line on a grid. *Computer Graphics and Image Processing* **3** (No. 1), 48–62 (March 1974).
9. H. Freeman, Boundary encoding and processing in *Picture Processing and Psychopictorics*, ed. by B. S. Lipkin and A. Rosenfeld, pp. 241–266. Academic Press, New York (1970).
10. C. Arcelli and A. Massarotti, On the parallel generation of straight digital lines. *Computer Graphics and Image Processing* **7** (No. 1), 67–83 (February 1978).
11. J.E. Bresenham, *Incremental line compaction*. Technical Report TR29.0243. IBM System Communications Division, Research Triangle Park, North Carolina (January 1981).
12. A. Rosenfeld, Digital straight line segments. *IEEE Transactions Computers* **C–23 12**, 1264–1269 (December 1974).
13. J. Boothroyd and P. A. Hamilton, Exactly reversible plotter paths. *Australian Computer Journal* **2** (No. 1), 20–21 (1970).


Received February 1981



## BIBLIOGRAPHY


K. Belser, Comment on 'An improved algorithm for the generation of non-parametric curves'. *IEEE Transactions Computers* **C–25 1**, 103 (January 1976).

H. Freeman, On the encoding of arbitrary geometric configurations. *IRE Trans. EC–10* **2**, 260–268 (June 1961).

M. D. Gibbs, Angled vector generator program: *IBM Technical Disclosure Bulletin* **21** (No. 5), 2041–2044 (October 1978).

L. Gilman and A. J. Rose, *APL An Interactive Approach*. John Wiley and Sons, New York (1974).

S. K. Hoo, Accelerated Bresenham algorithm. *IBM Technical Disclosure Bulletin* **18** (No. 4), 1075–1077 (September 1975).

K. E. Iverson, *A programming language*, p. 12. John Wiley & Sons, Inc., New York (1962).

B. W. Jordan, W. J. Lennon and B. C. Holm, An improved algorithm for the generation of non-parametric curves: *IEEE Trans. Computers* **C–22** (No. 12), pp. 1052–1060 (December 1973).

M. L. V. Pitteway, Algorithm for drawing elipses or hyperbolae with a digital plotter. *The Computer Journal* **10** (No. 3), 282–289 (November 1967).

J. Ramot, Non-parametric curves. *IEEE Transactions Computers* **C–25** (No. 1), 103–104 (January 1976).

F. Rubin, Generation of non-parametric curves. *IEEE Transactions Computers* **C–25, 1**, 103 (January 1976).

P. Coueignoux and R. Guedj, Computer generation of colored planer patterns on TV-like rasters. *Proceedings of the IEEE* **68** (No. 7) 909–922 (July 1980).