FONT VARIATIONS IN VECTOR PLOTTER LETTERING

Kurt E. Brassel and Jack J. Utano
Geographic Information Systems Laboratory
State University of New York at Buffalo
Amherst, New York 14260

Introduction

Pen-and ink vector plotters have recently become standard equipment in most computer installations. These drum or flatbed hardware devices are driven most often by a standard plotting software package supported by the hardware vendor (see Dunn and Herzog, 1977). Such software provides for the usual array of point symbol and vector mode line drawing but contains only a single type font for graphic character display. This lettering style is generally a standard, regimented one (usually a hybrid of Simplex Roman) and is appropriate for crude work graphs of minor graphical quality. Browsing through scientific journals or observing messages and labels generated from these standard plotting packages testifies to the inadequacy of using this restrictive type font for effective text communication.

The graphic and communicative quality of a plot rests heavily on its annotation. The recognition of the attribute(s) to which the text applies, the ease with which the lettering can be read and interpreted, and the 'balance' it renders to the rest of the plot are all essential aspects of graph/map design (Robinson and Sale, 1969). Since the standard plotting systems provide only a letter size variation capability and do not furnish pen-stroke width and inter-character spacing features, they are not adequate software media for generating the proper text-to-plot relationship that is necessary for effective reader communication.

On the other hand, sophisticated lettering packages are currently on the market that offer a menu of lettering styles (see, Hershey, 1969; DISSPLA, 1970). The lettering system designed by A.V. Hershey, for example, is based on an extensive font collection where the characters are constructed by a succession of pen-strokes. Though the various styles are of superior quality for small font sizes, the Hershey package degenerates for plotting graphic characters of a height of approximately .7 inches and larger. At this threshold size, the successive pen-strokes become separated by white spaces and remain clearly visible and aesthetically unattractive. In addition, this package and most others of its kind require a special higher level language interface and considerable memory storage space (Wolcott and McCrackin, 1977).

This article presents the development of a highly transportable interface to the standard lettering software that is available on most vector plotting packages. This interface consists of a few small subroutines coded in a higher level language (FORTRAN IV) requiring a negligible amount of memory storage. The package allows for variations in character size, width, and weight, but does not include type face variations. The package has two operational versions; the first provides the capability of varying the width of a graphic symbol where no pen changes are required. This version is written in ANSI FORTRAN and is highly transportable to those installations supporting a FORTRAN compiler.

A second version, in addition to varying the pen-stroke width, offers an inter-character spacing feature. Moreover, an automatic spacing compensation is provided for such narrow letters as 'I', 'L', 'T', 'Y', etc. Its transportability, however, is somewhat restricted in that the version accesses single characters within the computer word storing the text by employing a shift function. Compilers that support an appropriate shifting function in FORTRAN should experience no real problem in implementing this version. Some of the font variations that may be derived from this package are displayed in Figure 1.

Although not yet implemented, a third version can be constructed for the evolving SIGGRAPH standardized graphical interface (see, Clark, 1977). Version one, character width variation, would be directly transportable. However, the inter-character spacing feature in the second version would pose major program modifications since the current SIGGRAPH graphics package contains an automatic space adjustment (the BASE parameter) in its symbol-plotting subroutine (SIGCHR).

Figure 1: Font Variation Examples; HT represents the height of the text string expressed in inches; WT is the character width as a percentage of the user-supplied text string height; and SP denotes the user-specified text string length in inches.

COMPUTER

ACM

FONT VARIATIONS

	HT	WT	SP
SIGGR	.7	25%	3.6
SIGGRA	.6	20%	3.6
SIGGRAP	.5	10%	3.6
SIGGRAPH	.4	15%	3.6
SIGGRAPH-AC	.3	20%	3.6
SIGGRAPH-ACM 1977	.2	25%	3.6
SIGGRAPH-ACM A QUARTERLY REPORT	.1	10%	3.6
SIGGRAPH-ACM A QUARTERLY REPORT	.1	0%	3.6

GRAPHICS

Version One: Variation in the Pen-Stroke Width

In a vector mode environment, a graphic character is constructed by drawing a sequence of straight line segments, each segment being delimited by two endpoints or vertices. Within this context the line segment endpoints may be distinguished between vertices connecting two segments (inflection vertices) and vertices at the extreme points of a single segment (end-stroke vertices). In exceptional cases a graphic character may consist of a single stroke, as opposed to the more general multi-stroke characters.

Variation in the pen-stroke width of a graphic character may be achieved by the repetitive drawing of the character in a systematic arrangement. As shown in Figure 2a, the distance between two analog pen-strokes in a displaced character varies as a function of the displacement direction (in X and/or Y) and the stroke direction. Linear displacement, however, results in varying stroke widths for all multi-stroke characters and in some instances this may not be desirable. For example, a repetitive displacement of a multi-stroke character in the horizontal direction by a single pen width produces a character displaying uneven stroke widths similar to a Simplex Roman style (Figure 2b). The same would also apply for a character displaced in an arbitrary direction. As shown in Figure 3a uneven stroke widths may in many cases be undesirable.

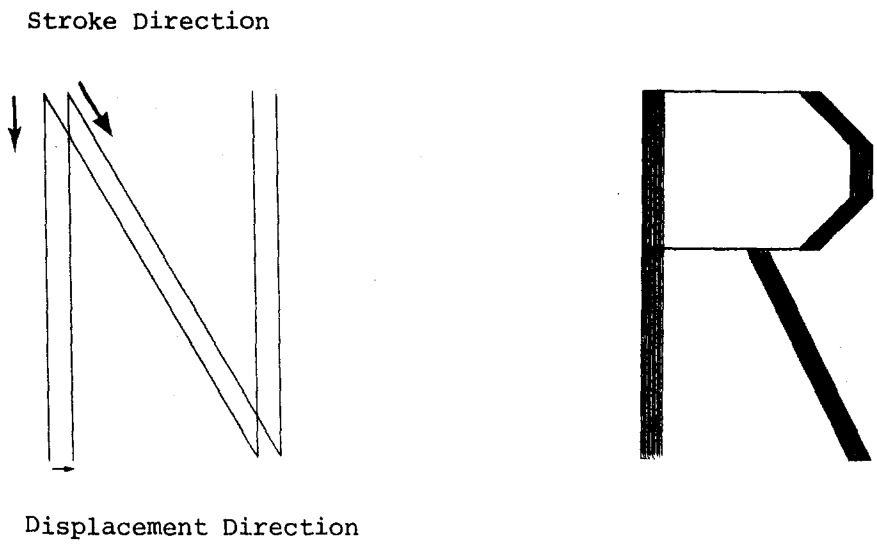
To avoid uneven strokes, a character is represented here via a circular displacement pattern. Given user-supplied text string height and width parameters, the number of repetitive characters to be placed around a circle is determined and the starting vertices are positioned at constant intervals on the circumference of the displacement circle. The number of repetitions to represent a character is chosen so that the chord between two neighboring points on the circle is approximately equal to the user-supplied pen width. This number is determined by the following equation:

$$NV = \frac{2\pi * RAD}{PW} = \frac{\pi * H * W}{PW}$$

where,

NV = the number of vertices to position around the circumference, to be rounded to the next higher odd integer
PW = the user-supplied pen width
H = the user-supplied text string height
W = the user-supplied character width
RAD = the radius of the circle for the character, determined by:

$$RAD = H * W/2$$



a b

Figure 2.
a: Plotting graphic symbols via a Displacement Direction and Stroke Direction.

b: Plotting a character through a horizontal displacement.

a b

Figure 3.
a: Example of a horizontal displacement of a character resulting in uneven stroke widths.

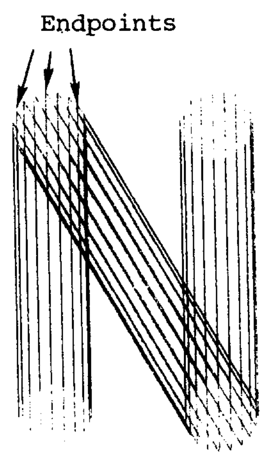
b: Graphic symbol representation via a circular pen-stroke displacement scheme.

According to the scheme, pen-stroke displacement produces a character bearing a consistent pen-stroke width, and the width of the character is equal to the diameter of the displacement circle (Figure 3b). The character height in this version is equal to the sum of the user-supplied height H and width W. Thus, it is recommended that the user define the character height as H - W to get a symbol height of H. Each of the pen-stroke endpoints are of a circular shape and the spacing of the repetitive lines connecting the endpoints are proportional to the difference of the sine functions of the angles between the stroke direction and the connection of the stroke endpoint with the center of the circle (Figures 4a and 4b). Plotting overlap, however, is unavoidable for multi-stroke characters. The overlap occurs at each inflection vertex in the symbol (Figure 4a).

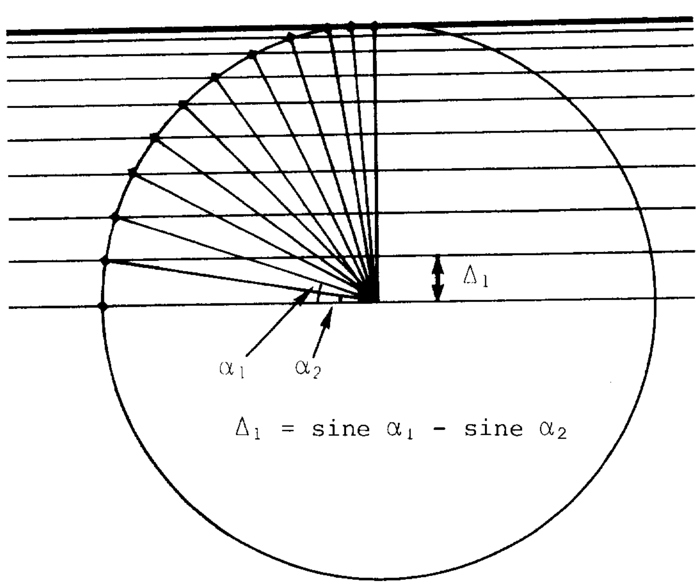
Version Two: Character Spacing and Inter-Character Space Adjustment

Standard lettering packages usually employ a square matrix arrangement for character placement where the character width (character image plus intercharacter space) is equivalent to the character height. The relationship between the character image and intercharacter space is generally 60%:40% of the character height. Narrow characters like 'I', 'l', ';' and ':' etc. have relationships as indicated in Figure 5. If a text string is to be plotted, however, using Version One where a pen-stroke width parameter is supplied, the 60:40 relationship is forfeited. Empirical testing has shown that with a character width of 30% or more of the specified height, the plotted characters become 'over-crowded' and begin to coalesce.

Version Two, however, may be used to compensate for a pen-stroke width displacement to preserve the 60:40 square matrix relationship. Given the user-supplied pen-stroke width, a new height:width relationship is established. This relationship for a character height H and pen-stroke width W are described in Figure 5 for regular characters and for narrow characters. Here, the actual size of any given character is found by H' = H - W and the overall character height is achieved through the repetitive plotting of the character in the circular displacement scheme in Version One. Narrow characters are buttressed with an extra spacing of 15% on each side to accommodate the normal spacing adjustment. Spacing compensation is also given to open letters like 'T', 'Y', 'P', 'F', and 'J'. In some cases this compensation is given in only one direction (either forward or backward depending on the particular character), and is omitted for



a



b

Figure 4.

a: Pen-stroke endpoints around the displacement circle.

b: Repetitive line spacing; where,

α_i = the angle between two adjacent vertex positions and the center of the displacement circle.

Δ_i = the distance (i.e. the space) between the placement of successive lines.

the first and last characters in the text string.

Version Two contains an alternative character spacing arrangement. Given the user-supplied height, pen-stroke width, and overall text string width, the cumulative spacing required for the string is computed and the complement of the overall text string width is divided into equal intercharacter spaces. The string is then plotted within the bounds of this user-defined rectangle.

<u>Access to the Lettering Interfaces</u>

The packages have been implemented and incorporated into the CalComp plotting software environment where the calls to the operational routines are aligned with the CalComp plotter calls (see, California Computer Products, Inc., 1976). Access to the first version is provided by the following call:

```
CALL SYMBL (X,Y,HEIGHT,BCD,ANG,INT,PENW,WDTH)
```

where the first five parameters are identical to those in the CalComp routine SYMBOL. These are:

X,Y	Lower left hand corner of the text string
HEIGHT	Character height of the text string
BCD	Character string to be represented
ANG	Angular direction of the text string
INT	Number of characters to plot.

In addition to the CalComp variables, the following two parameters are necessary:

PENW	width of the pen to use (usually from ball-point or liquid ink)
WDTH	width of the character(s) in the text string -- expressed as a percentage of the user-supplied height.

Since the CalComp package contains a NUMBER plotter call for numerical values, the following call is used for Version One of the interface:

```
CALL NUMBR (X,Y,HEIGHT,FNUM,ANG,INT,PENW,WDTH)
```

where the call parameters are identical to those in SYMBL, except for FNUM which corresponds to the numerical value to be represented and INT which denotes the number of decimal digits to plot. Figure 6a displays a sample of SYMBL and NUMBR of Version One.

The second version has the following access format:

```
CALL SYMB (X,Y,HEIGHT,BCD,ANG,INT,PENW,WDTH,SPACE)
```

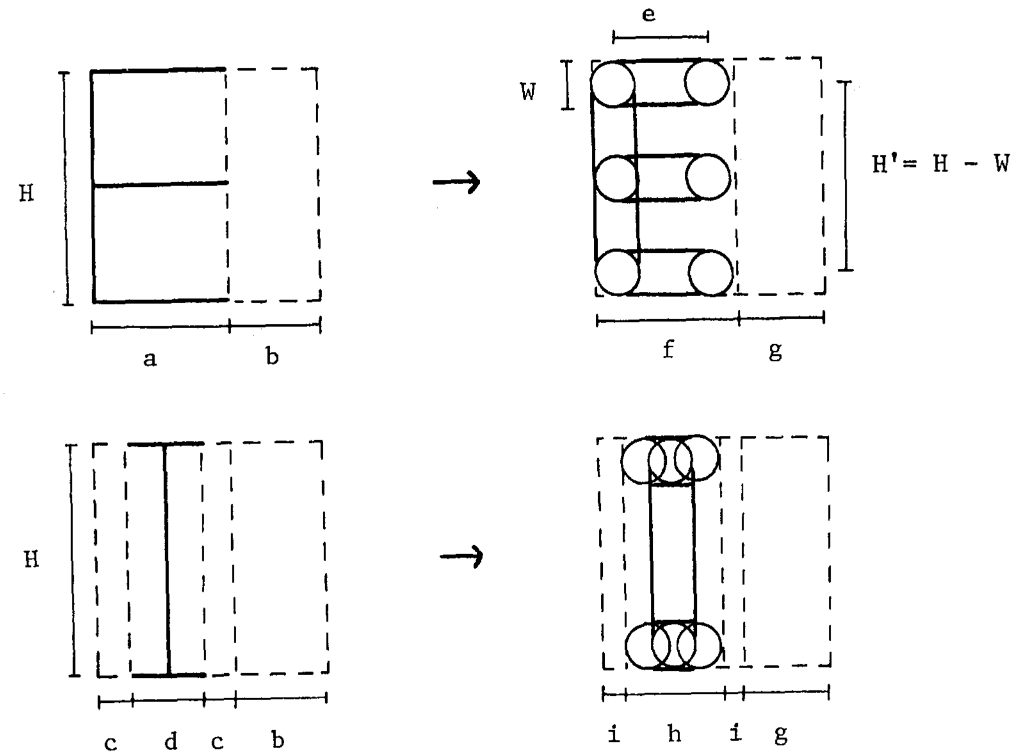


Figure 5. Symbol placement dimensions in standard and narrow characters; where,

H = user-supplied character height
W = user-supplied character width

a = .6H	d = .3H	g = .4(H-W)
b = .4H	e = .6(H-W)	h = .3H+.7W
c = .15H	f = .6H+.4W	i = .15(H-W)

VERSION ONE 1977

a

V E R S I O N T W O 1 9 7 7

b

Figure 6.

a: CALL SYMBL(X,Y,.285,11HVERSION ONE,.0,11,.013,.15)
CALL NUMBR(X,Y,.285,1977.,.0,0,.013,.15)

b: CALL SYMB(X,Y,.3,11HVERSION TWO,.0,11,.013,.15,3.8)
CALL NUMB(X,Y,.3,1977.,.0,0,.013,.15,1.4)

CALL NUMB (X,Y,HEIGHT,FNUM,ANG,INT,PENW,WDTH,SPACE)

here the first seven parameters correspond to those detailed in Version One. The additional SPACE parameter is used to designate one of the two spacing options:

SPACE (negative value): the absolute value is used to determine the relative intercharacter spacing for the text string. It specifies the spacing to be a function of the text string height (e.g. SPACE = -.10, intercharacter spacing is 10% of the text string height).

SPACE (positive value): specifies the overall width of the text string. Together with the Height parameter, a rectangular box is determined in which the text string is to be entirely positioned. The appropriate intercharacter spacing is then established (e.g. SPACE = 6.0, the text string is placed to fit within a box six inches wide).

Figure 6 displays a sample of SYMB and NUMB for Version Two.

Final Comments

A simple and inexpensive lettering interface package has been developed (and implemented on a CalComp Model 925/936 Plotting System) that permits variation of the character height, pen-stroke width, and character space adjustment foı special characters. The lettering interface can be incorporated into any pen-and-ink vector plotter hardware/software environment. Version One may be transferred to any FORTRAN compiler without modification, whereas the second version requires the existence of an appropriate FORTRAN shift function for successful transportability.

It must be noted, however, that our algorithm for plotting characters has one weakness; repetitive character plotting may result in a multiplication of processing and plotting time. From this point of view, it is recommended that 'heavy' pen widths be used for the large headline-type lettering. Taking this point into consideration the proposed interface provides for a versatile and inexpensive tool to produce a broad range of lettering of considerable graphical quality.

REFERENCES

Ali, Moonis, 1977, "Mathematical Picture Grammer Applied to Script Generation," Computer Graphics and Image Processing, 6, 93-102.

Berger, C., 1944, "I. Stroke Width, Form and Horizontal Spacing of Numerals as Determinants of the Threshold of Recognition," Journal of Applied Psychology, 28.

Bridgeman, C.S. and E.A. Wade, 1956, "Optimum Letter Size For a Given Display Area," Journal of Applied Psychology, 40, 378-81.

CalComp Software Reference Manual, October 1976, California Computer Products Inc., Anaheim, California.

Clark, Robert, 1977, "Graphics Algorithms," Computer Graphics, 11, #1, 47-51.

DISSPLA Advanced Manual, 1970, Integrated Software Systems Corp., San Diego, California.

Dunn, Robert M. and Bertran Herzog (ed.), 1977, "Status Report of the Graphics Standards Planning Committee of ACM/SIGGRAPH," Computer Graphics, 11, #3, part I.

Hershey, A.V., 1969, FORTRAN IV Programming for Cartography and Typography, U.S. Naval Weapons Laboratory Technical Report TR-2339, Dahlgren, Virginia.

Johnson, Clyde G., A.V. Hershey and A.L. LeBlanc, 1974, "Cartographic Symbology Panel," Proceedings of the International Conference on Automation in Cartography, ACSM/USGS, Reston, Virginia, 215-239.

Keates, J.S., 1973, Cartographic Design and Production, Halsted Press, Wiley & Sons, New York, 240 pgs.

Kernighan, B.W. and L.L. Cherry, 1975, "A System for Typesetting Mathematics," Communication of the ACM, 18, #3, 151-157.

Kriloff, Harvey Z., 1976, "Human Factor Considerations for Interactive Display Systems," User-Oriented Design of Interactive Graphics Systems, ACM/SIGGRAPH Workshop, Pittsburgh, Pennsylvania, 45-52.

Robinson, Arthur and Randall D. Sale, 1969, Elements of Cartography, John Wiley and Sons, New York.

Wilkie, William T., 1973, Computerized Cartographic Name Processing, M.S. Thesis, University of Saskatchewan, Saskatoon, Saskatchewan.

Wolcott, N.M. and F.L. McCrackin, 1977, "A FORTRAN Program to Draw Enhanced Graphic Characters," Computer Graphics, SIGGRAPH '77 Proceedings, 11, #2, 121-127.