April 1983

Report No. STAN-CS-83-971

Letterform Design Systems

by

Lynn Ruggles

Department of Computer Science

Stanford University
Stanford, CA 94305

Report No. STAN-CS-83-971
April 1983

Letterforrn Design Systems

Lynn Ruggles

Department of Computer Science
Stan ford University

The preparation of this report was supported in part by the National Science Foundation under grant IST-8201926 and by the System Dcvclopment Foundation.

Electronics will soon force its claims upon letterforms, and let us hope it will liberate us from the dust of the past.
Hermann Zapf, 1968

Abstract

The design of letterforms requires a skilled hand, an eye for fine detail and an understanding of the letterforms themselves. This work has traditionally been done by experienced artisans, but in the last fifteen years there have been attempts to integrate the design process with the use of computers in order to create digital type forms. The use of design systems for the creation of these digital forms has led to an analysis of the way type designs are created by type designers. Their methods have been integrated into a variety of systems for creating digital forms. This paper describes these design systems and discusses the relevant issues for the success of the systems that exist and are used today.

--

Introduction

In recent years, the availability of extremely fast processors and high resolution graphics devices such as display screens and dot-matrix printers has led to an investigation into the way text is manipulated and stored. Methods of encoding digital information and the speed at which it can be created and modified have contributed to the increased use of digital information in place of traditionally analog forms. One area that is under careful scrutiny and analysis is the field of digital grammatology or letterform design. The conversion of an artistic design into a process employing computers and digital output has only been partially successful thus far, although work is continuing on the integration of computer technology with' the conceptual design process.

A typeface or letterform that has been stored as digital information is referred to as digital type. This information is created by converting a black and white image into a matrix of black and white dots or pixels (picture elements). The matrix is called a raster image or bitmap and can be used by phototypesetters, CRT displays, laser typesetters, and other output devices capable of utilizing digitized images.

The letterform design systems surveyed here all have a minimum of three components: an input mechanism, a means of interacting with the system, and a method of displaying output from the system. The input to the system may come from a variety of sources including cards, tape, an optical scanner, a tablet and electronic sensor, menu selection, a lightpen, screen and keyboard, or a description of the image such as a program. The system may allow USC of any of these devices or may be constrained to use only one or two.

The second component, the user interface, can be compared to the dialog between the system and the designer. A friendly system will converse in a language that a designer can easily learn and understand, whereas a system that has not been written with the interface as a major concern may be complex, fmstrating and not easily learned or used. The level at which the designer interacts with the system is an additional consideration when evaluating the interface. The system may provide facilities for the initial design of a letter and then allow modifications to it to produce a finished product, or it may restrict the input to an encoding of an existing letterform and limit operations to a small set of editing commands. Visual interaction allows display and editing of a letterform on a video screen in an interactive environment while a system with a batch type of processing accepts changes to the letterforms in batches then processes them all at once.

Storage of data in the system also affects interaction with the system. For example, a *library facility* can provide a means for storing either entire letterforms or parts of letters for later use. Items in the library can be stored either as raster images or as descriptions of the images. Sometimes, the saved images can be manipulated by actions such as rotation, reflection, or stretching in one or several directions.

The last component, 'the method of displaying- output from the system, determines whether the letterforms are printed on paper or film or saved as digital information for use by display devices such as printers or video screens. The output medium is of crucial importance as it determines whether the letterforms are produced at a very fine resolution with the details of the letterforms readily apparent, or whether the printed image is only a crude approximation to the actual letterform design. The design system, despite all the wonderful features which it may have, is limited by the output from the system. A system which can produce high quality letterforms, but cannot display or print them at that high a quality, is not of much use.

Conversion of a letterform design into digital information can be a slow, laborious process. A typical sequence of events carried out by a letterform designer* would be to first prepare an original character drawing, which is then scanned by an optical scanner. The scanner passes a beam of light repeatedly over the image, with each pass covering only a small band of the image. The dark parts of the letter are converted into dark pixels, the light parts into white ones. The scanning width of the beam can bc very small, sometimes as fine a few microns in width, thus producing a very high resolution raster image. Problems associated with this method include dropouts and pickups, or small indentations or bumps on the interior and exterior edges of the digital character. These imperfections are caused by scratches, dirt, dust and blurred edges, all of which contribute noise to the scanner and lead to degradation of the digital data. An additional problem

*In this paper, in order to distinguish between the designer of a lctterform design system and a lctterform designer using such a system, the term author will bc used to refer to the person or persons who developed the system and the term designer will describe a person using the system. .

is a staircase effect along the edge of a letterform, a phenomenon also called the 'jaggies.' This problem occurs on any edge that is not aligned exactly vertically or horizontally, such as curves or diagonals, and also affects lines which should be parallel or perpendicular to the baseline but are not lined up exactly with the direction of the scanner beam.

The dot matrix image of the character resulting from the scanning process is then proofed and corrected. To facilitate this process, the image is printed in both a large and small size. The small image is inspected and corrections made to the large one. This step is crucial and is done by trained letter drawing specialists on either a graphics display screen or on a printed copy of the raster image. Any corrections to the digital form are then entered into the system.

The design process also includes determination of the correct spacing between the letterforms. Combinations of characters are examined and adjustments are made so that groups of characters appear evenly spaced. The process of correcting the image and recording the corrections is repeated until the letterforms and their spacing are satisfactory.

Once a design has been created, it may be used to produce several different masters or sizes by enlarging or shrinking the digital image. Modification of a letterform by scaling results in distortions of characteristics of the form such as the serifs (the finishing strokes on a letter), the hairlines (the thin parts of the strokes), and the stem widths (the main vertical strokes in the letter). Some features on the resulting masters may be too light in the smaller sizes and too heavy in the larger ones. If the distortions are very bad, each of the new images must be hand edited.

In most of the systems surveyed, letterforms have been encoded through one of two methods. Both require that the letterform image be marked with control points along the interior and exterior edges of the character. These points are then entered into the system by specifying the coordinates of the points, either explicitly through USC of an auxillary device which can point to each of the control points, or through a list containing the coordinates of the points which is entered either as data or a program.

Systems using the first method of encoding employ a digitizer tablet with an electronic sensor or puck containing a set of selector buttons and a window enclosing two crossed wires. The marked character is laid on the tablet and the points are entered by specifying their coordinates through use of the puck. The crossed wires, or crosshairs, are lined up with each control point and one of the selector buttons is pressed. The coordinates of the point are determined by an electronic grid in the tablet which senses the position of the crosshairs. These coordinates are saved as is information about which button was selected. Different buttons indicate different types of edge points such as curves, corners, starting points or tangents. Use of the tablet and puck is faster for the input of images but the resulting forms still have to be edited because of distortions to the forms when they are converted to different sizes.

The second method of entering points uses either a program or a fixed set

of data to describe or list the coordinates of the control points. Additionally, instructions are given specifying how points are to be connected together through the use of straight lines and curves such as splines. The shape-of the character at different sizes and resolutions can be controlled by modifiying the description of the parts that change. Other parts which make up the letterforms are not affected. Modifications to the parts that are affected by a change in size can be controlled by the size of the font.

For further discussion of digital type and the problems associated with it see Bigelow [3,4,5] and Seybold [20].

Early Work

In a paper entitled *Three Fonts of Computer-drawn Letters*, published in 1967, M. V. Mathews, Carol Lochbaum, and Judith A. Moss describe three groups of letters drawn with a vector display on a cathode ray tube [14]. Input to their system was on cards and processing was done in batch. The edges of the letter were described using short vectors and the images were displayed on a cathode ray tube. To select the edges of the letter, a picture of the letter was laid on a grid, and points were selected visually along the edges. The coordinates of the points were punched on cards and input to the system. To output a letter, the points were connected to form vectors which were then drawn on the screen of the output device. The fuzziness of the display medium and the low resolution of the device contributed to round out the straight lines making up the letter to give a smooth edge (figure 1). Although the input method was tedious, the authors were enthusiastic about both the quality of the output and the ease with which new shapes could be created and saved for later use.

In the same year, Allen V. Hershey published a paper, *Calligraphy for Computers,* describing work he did using computers and CRT printers for preparation of mathematical reports [8,9]. Hc created a large repertory of digitized characters corresponding to the repertories of the American Institute of Physics and the American Mathematical Society. The characters that he created were displayed on a vector plotter and were made up of points connected by straight line segments. Data for each character consisted of x,y coordinate pairs, the first pair giving the starting point of the letter and each successive one giving the displacement from the previous point. This data was punched on cards, then read in and stored on tape for later USC. Output was on a dot plotter.

Hershey was concerned with the quality of the letters that were output by his system and felt that the letters that he designed were a compromise between smallness, smoothness and legibility. He thought that if the letters were too small, the CRT beam would bridge small gaps or fill in small openings in the characters thus leading to decreased legibility. He wanted, however, to be able to print as many characters on a line as possible and for the letterforms to have a smooth, professional appearancc. Because of his concern for the appearance of the letters,

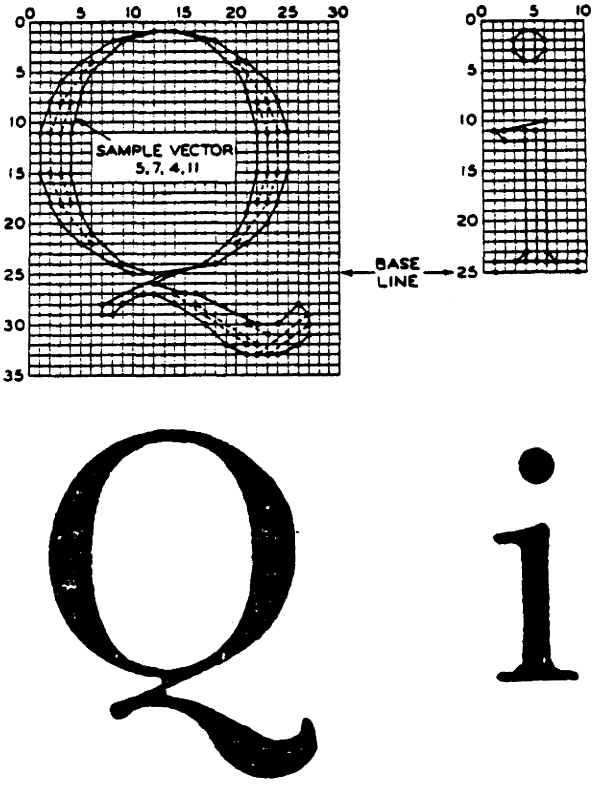

Figure 1. Letters drawn with vectors on a CRT screen.

he attempted to match the widths and heights of the letters to an integer number of raster units to avoid rounding errors. He also designed uniform width characters so that any letter could be used interchangeably with another and consideration would not have to be given to letter spacing. His letters were designed in one size only but did include a large variety of fonts including types he called Roman, Greek, Italic, Russian, Script, three styles of Gothic, named Italian, German, and English, music notation, and over 6000 Japanese characters (figure 2).

ITSLF

ITSLF, or InTeractive Synthesizer of LetterForms, was designed by H. W. Mergler and P. M. Vargo in 1967 [17]. The design of ITSLF was an attempt to merge the USC of photocomposition in the printing industry with the development of computer graphics and computer assisted design systems. In addition, the data manipulation capabilities of computers were used to assist the efforts of the typographic designer. The computer performed all of the quantitative work such as maintaining stem widths, height, and depth, thus freeing the type designer to work on the characteristics of the design.

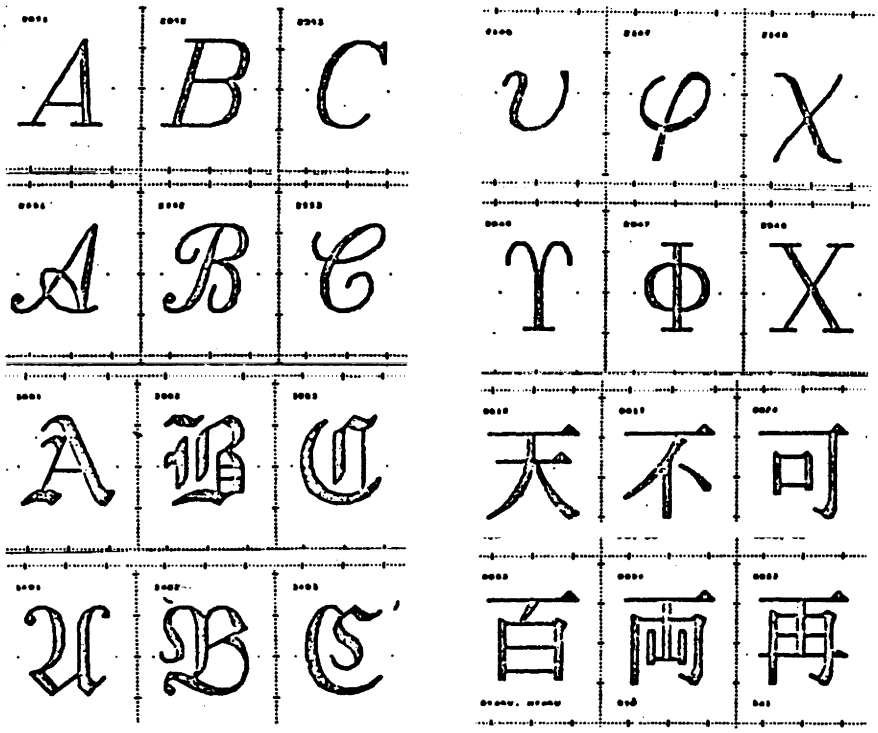

Figure 2. A sampling of Hershey's characters.

Mergler and Vargo were influenced by the observation by Fredrick Goudy, a noted type designer, that it was easy to design one letter, but difficult to design the rest of the alphabet to harmonize with that letter. They assumed that typographic designers only needed to make choices on the desirability of certain characteristics in a typeface design such as the letterform shapes, heights, widths, stroke weights and serif shapes in order to produce different typeface designs. These characteristics could bc generalized to a set of parameters which incorporated the essential characteristics of a design so that modification of these parameters was the only step necessary to produce different designs. Limiting the designer to specifying certain parameters instead of the entire design of each letter would lead to consistency in the overall design of the characters.

As a first step in the evolution of ITSLF, the alphabet was divided into classes based on gcomctric propertics of the lcttcrs. The gcomctric characteristics of the letters in the alphabet were extracted and saved as a combination of straight lines, curve strokes based on the supcrcllipsc, and tails. A set of criteria in the design was determined such as the height, weight, and stem width which were then used as parameters and adjusted to provide different designs. Additionally, two types of serifs were dcfincd, one for horizontal strokes and one for vertical or slanted strokes. An operator of the system cntcrcd values for the parameters which ITSLF then used to modify its geometric data to produce drawings of the letters;

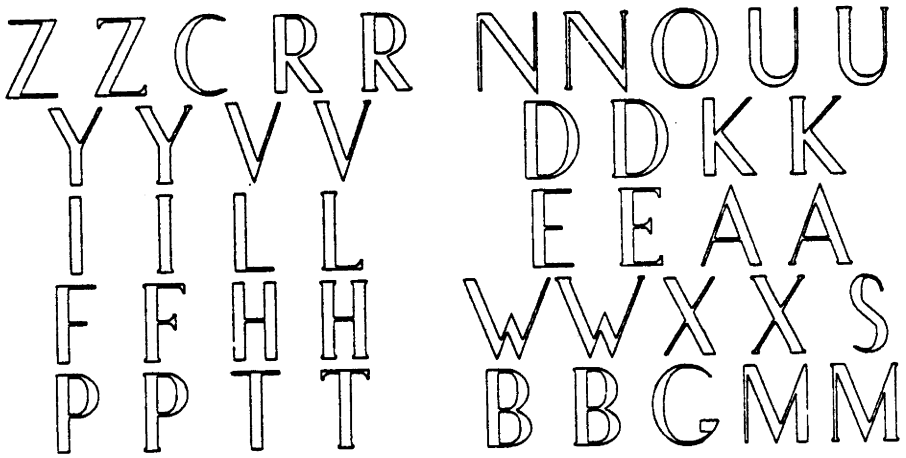

Figure 3. Serif and sax-t-serif letters generated by ITSLF in automatic mode.

ITSLF was restricted to the generation of only 24 roman capital letters. The 'Q' and the 'J' were left out because they were so similar to the 'O' and the 'I'. The system had two modes, manual and automatic. In manual mode, the operator could generate letters that were dissimilar to each other by specifying different parameter values for each letter. In automatic mode, the operator entered only the values for the parameters for the letter 'E' from which the paramcters for the other 23 letters were calculated. Automatic mode thus guaranteed a consistant set of letters (figure 3).

The data used to determine the geometrical properties of the letters and the significant parameters came from 18-24 point letters photographically enlarged 32 times. Output from the system was on a Cal-Comp plotter at either four times or thirty-two times actual size.

Although ITSLF never became an actual production system, it was one of the earliest systems designed specifically for typefaces. Despite its limitations, Mergler and Vargo felt that the basic premise of their design was valid and that there existed a set of parameters which could be used to control a geometric design. They did feel, however, that much more study of the characters was necessary to fully benefit from the potential of their program.

CSD, Character Simulated Design

The idea of characterizing an alphabet by generating. descriptions of the characters and then modifying those descriptions by the use of paramcters was also used by Phillippe Coueignoux in 1975 in his Ph. D. dissertation, *Generation of Roman Printed Fonts* [7]. His work included CSD, or Character Simulated Design, a program to gcncratc high quality digital characters. Although it appears to bc an

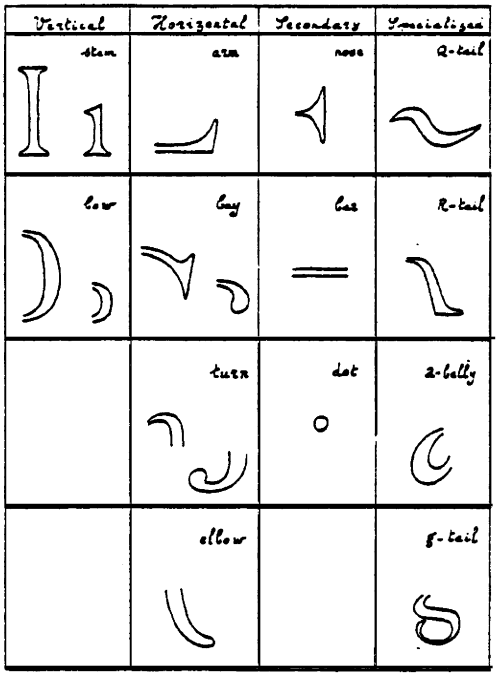

Figure 4. Primitives used by CSD

extension of Mergler and Vargo's ideas, Coueignoux did not know about ITSLF until after his own work was completed.

Coueignoux's work was derived from an attempt to draw fonts with a computer. He studied the structure of the characters from a generative point of view, and thus wanted to quantify or extract the consistency in Roman type fonts both between one letter in different fonts and between all letters in one font. He derived primitives such as stems, arms, and noses and defined the spatial relationships between them. From careful evaluation of these primitives, he generated a grammar to describe the implicit structure of the characters in a font. This grammar included two rules: *the rule of proportion* specified the relationship between the values for the parameters, and the *rule of disposition* described how the primitives were combined within a font. The grammar included rules for the links between primitives, for specifying the stem thicknesses of the primitives and their position, length, and truncation, and for describing parts of the characters and the constraints on them. A rule specified the relationships between the parts and described new parts from existing ones. The set of primitives is shown in figure 4.

The parameters in his system were used to specify the likeness between the occurrences of a primitive within a type font. They were originally determined by measuring a set of letters by hand with a ruler and were then entered into the system. There were an average of ten parameters per primitive, with the actual number varying from 3 to 30. The parameters included the height, thick and thin

thicknesses, horizontal extension, angle, and squareness. The curves used in CSD were made up of conic sections, straight lines, or bows (superellipses).

CSD contained a set of routines, one for each letter. To output a letter, the corresponding routine was given parameter values for the primitives in the letter. The system actually only had 44 routines since some routines were used for both the upper and lower case characters, the latter simply being a scaled down version of the former. The system was oriented toward creation of letterforms, and for that reason only stored the descriptions of the primitive parameters and did not save any infonnation about the letters that were created. Every time the system was used, the values of the parameters for each letter had to be reentered. When a letter was designed, it was displayed on a screen and the parameters could be adjusted interactively. When a final proof was desired, the characters were output on a 200 dot per inch electrostatic plotter. This output was then photographed and reduced.

U S S R

In October 1977, A. A. Bers at the Academy of Science in the Soviet Union wrote a paper, *Implementation and Design for the Printing of Typefaces on an Automatic Phototypesetting System* [2]. His program read in a bitmap character representation and generated an outline contour using straight and diagonal lines and four different kinds of curves. This information could then be used to create roman, italic or cursive forms, with light, regular, meduim bold, or bold stem weights, and the character spacing could be specified as extra condensed, condensed, normal, extended, or extra extended. Some sample characters are shown in figure 5.

ELF

ELF is a letter design system developed by David Kindersley and Neil Wiseman around 1978 which is still in use [12,25]. It includes an interactive display with a keyboard and lightpen and is used for the creation, manipulation, measurement or copying of images. Within this system, the design of a typeface is supported from its initial stage of sketches of letterforms to the final one of encoding the finished forms into fonts.

The display mechanism allows one active and three passive images to be displayed on the screen at one time. The active image is the one that is currently being edited. There are two cursors associated with the active image: the *point cursor* surrounds a selected point, the *line cursor* is positioned at the mid-point of a selected line (figure 6a). The selected line terminates on the selected point. Each point is specified by four values: its x and y coordinates, its amplitude, called z, and its style which refers to the type of line which terminates at the point. The style can be horizontal, vertical, any orientation, or invisible.

Figure 5. Output from Bers' work.

To design or draw a character, a designer moves a lightpen over the surface of a screen creating a sequence of line segments. This movement is interpreted by the system as if it were a pen of a certain width moving across the screen. The line segments, or line chain, is replaced by a series of trapezoidal shapes whose baseline is perpendicular to the x or y axis (figure 6b). The baseline is 100 units wide and is analogous to a pen with a width of 100 units and a slope of 0. This width, or amplitude, can be modified by the designer in order to produce lines with different thicknesses. The angle. of the pen can also be changed.

After the design is input to the system, the character must be edited through menu selection and an interactive language containing 30 commands. Commands can change the style of the selected line, circulate the cursor forward or backward, calculate a filled area version of a picture and display it, delete points, insert points, alter the x,y, or z values for any point, join a copy of a passive image to the active image, or change the value of a variety of parameters. These parameters control the responsiveness and sampling rate of the light pen, the behavior of the fill command, or the type of picture displayed.

The system maintains both an internal model with details recorded so they can be easily manipulated by geometric transformations, and a log, in 3 textual

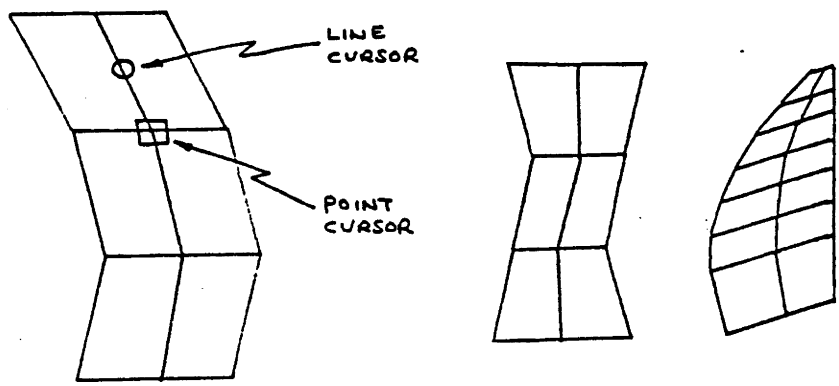

Figure 6. Line and point representation used by ELF.

form, of the sequence of actions. The log file can be edited before being used to replay an earlier design session or to recover a previous state. Output from the system is on a laser display plotter.

A few commands are provided specifically for designing letters. For example, one such command allows several characters to be displayed together on the screen allowing for judgement of the spacing of the characters as the letters are designed. The system also includes curve generators and interpolation control.

PM Digital Spiral

The PM Digital Spiral developed by Peter Purdy and Ronald McIntosh is a bitmap editor system based on the ideas that the outline of the character is its definitive aspect and distortions along the outline are less tolerable than those inside it [19]. They recognized that a digital image is greatly improved when enclosed by a smooth edge, so they decided to provide a means to do this. Their system allows an designer to trace around digitized letters using a computer-stored spiral which was generated on an extremely high resolution screen. The designer uses the spiral to trace around the outline of the letter, smoothing away the irregularities such as dropouts and pickups inherent in the digitizing process. After the edges of the letter have been outlined with the spiral, the coordinates of the outline are stored and later used to recreate the image. The combination of the spiral with the high resolution screen gives a very sharp edge.

The spiral, also called the Typographer's Curve or Template, is a two convolution spiral (figure 7b). It is made up of 16,384 (2^{14}) steps each oriented in a North, South, East or West direction. Each of these steps is randomly addressable. The spiral is used by moving it on the display screen to an edge of a displayed bitmapped character and then uncurling it until a curve on the spiral is found to match the desired curve of the edge. The endpoints of the curve are then specified

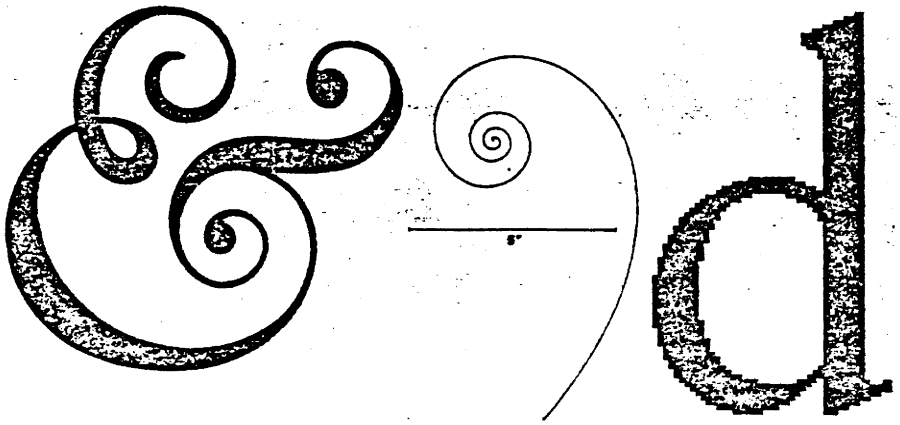

Figure 7. a) Original design of a character with a smooth edge.
b) Typographer's Template c) Digitized character.

(figure 8). Each letter is thus made up of a small set of curve sections, or links, each specified by its length, angle and starting point, as well as whether it has to be invisible, reflected, or reversed. A straight line can be selected by specifying a curve with starting position zero. The set of links making up a letter is called a recipe.

In designing the PM Spiral system, Purdy and McIntosh recognized that type at different sizes must be proportioned differently, and that the ascenders and descenders must be proportioned when changing type sizes. Therefore, they provide an operation which they call *sidestepping.* Use of this operation forces the tracing beam of the video display to keep a measured distance from the recorded course while maintaining a 90 degree angle to its tangent. This method allows the system to generate images which are slightly larger or smaller than the original image, thus providing different sizes of the character for different resolutions, provided that the sizes are within a certain range. If sidestepping is used to create an image too much larger or smaller than the original, gross distortions of the image creep in.

As a future modification to their system, Purdy and McIntosh hope to include a provision for allowing a slightly irregular contour around the edges of a character which would imitate the 'ink-squash' inherent in hot-metal letterpress work. This deliberate blurring of the image would avoid any intense sharpness being maintained through to the final proof, a quality which they feel disturbs the readability of the page.

Ikarus System

In the last few years several sophisticated design systems created specifically for the design of letterforms have emerged. One of these is the Ikarus System,

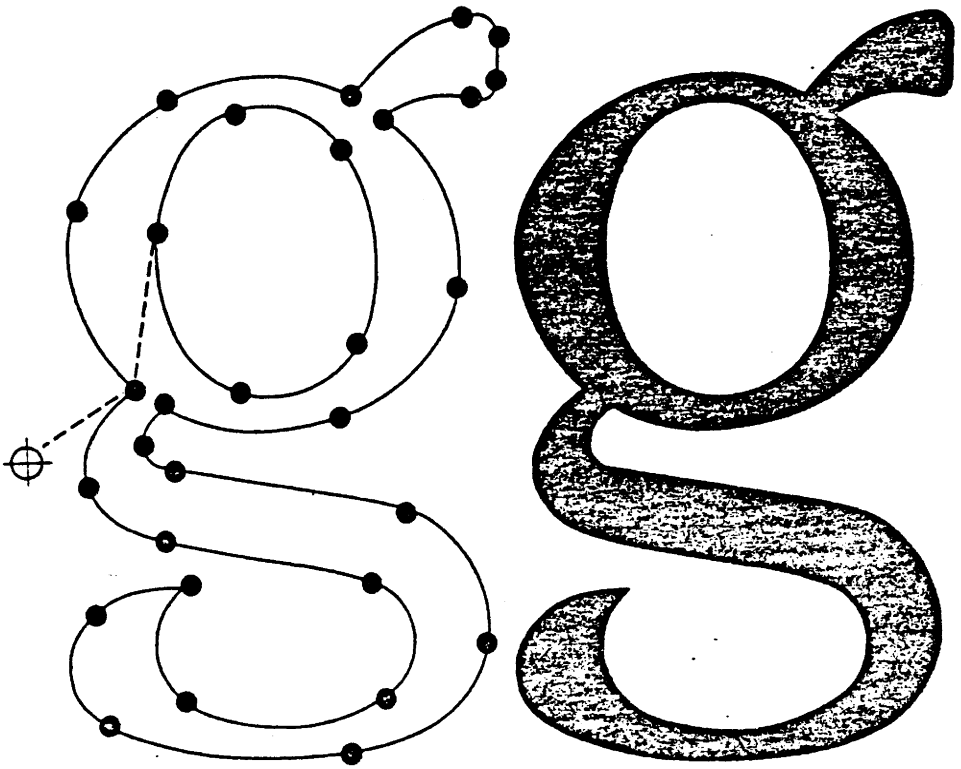

Figure 8. PM Edited Character.

developed by Peter Karow at the company Rubow Weber in 1973-1974 [10,11]. Ikarus provides a method of converting analog data such as letterforms into digital data.

As an initial step, the letterform must be marked for digitizing by specifying certain points along the edge of the character and whether each point is a starting, edge, curve or tangent point. The designer then uses a digitizer tablet and a puck to enter data. The puck is moved from point to point on the marked character. At each point, a button is pressed on the sensor. This action saves the x and y coordinates of the point and identifies the type of point. Additional information such as location and type of serifs can be specified via menu field selection or by keyboard input.

Ikarus allows adjustments to groups of characters with universal parameters such as height, width, horizontal and vertical alignment or slant, and whether the font is a serif or san-serif type. The x and y values can be changed separately thus allowing stretching or shrinking in one direction at a time. Modification can be made to characters individually without recalculation of an entire font. Editing of characters can be made either to the character displayed on a video screen via cursor or keyboard entries, or to a listing of the data representing a character. This

Figure 9. Variations provided by Ikarus a) Interpolation from light to bold. b) Variations in shading. c) Variations in rounding.

listing is in a format that is readable by a designer. Interpolation can be done from one type weight to another so that an designer need only design a light and a bold face and the system will provide the intermediate weights (figure 9a). The system can also provide rounded or shaded versions of a design (figures 9b,9c).

Ikarus also has the ability to convert data from digital images produced by an optical scanner to a contour or outline vector format. Ikarus can then perform several clean-up operations on the contour image to remove irregularities from the forms such as warts or bumps (dropouts or pickups) caused by images that had noise on the input (figure 10). *Channel-automization* can also be done on the contour data. This process justifies an image horizontally or vertically so that specified lines are parallel to the x and y axis even if the character's stem is not situated exactly perpendicular to the scan direction. Additionally, the system can match stem widths and serifs and adjust characters so that they fit within a required cap-height, x-height, or vertical position.

Output from Ikarus can be fit to a particular grid. If the stem width of a letter input to the system fits into 1.5 scan lines, the system has the ability to produce two versions: one with the stem only one scan line wide, the other with a stem two scan lines wide. This feature can circumvent the production of letters which have parts of varying widths.

The final results of the processing can be output to a variety of devices including CRT typesetting machines and drafting machines which cut the letters on film. The system can be interfaced to CAD/CAM peripherals such as digitizer tablets and precision plotters.

Metafont

Another sophisticated design system is METAFONT, designed by Donald

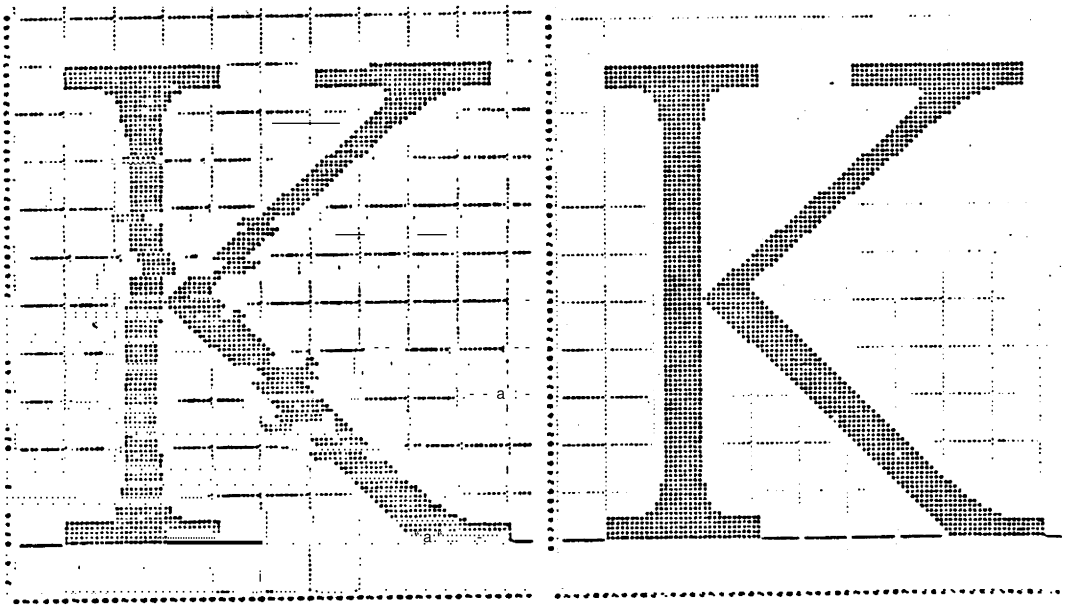

Figure IO. Clean up of digital character a) original data
b) data after input to Jkarus.

Knuth in 1979 [13]. Rather than constraining the system to a fixed set of letterform descriptions which arc modified by parameters, **METRFONT** allows the designer to design letters based on the movement of a pen across a surface such as a piece of paper or terminal screen. Thcsc letterforms are controlled by parameters, but it is the designer who defines both the images and the parameters that operate on them. The pen nib is specified to be any of a number of predefined shapes, or one created by thc designer. Additionally, the pen can be an eraser, allowing drawn lines to bc crased.

METAFONT is a programming language with which pen movements can be specified, much as in 'connect-the-dots' pictures. Points are specified by x and y coordinates and an optional tangent line. **METAFONT** programs comprise instructions listing which points should bc connected to which other points and in which order. The points can be connected by a wide pen, in which case the center of the pen passes through the points (figure lla). If thc pen has a very narrow width and the points lic on thc edgcs of a character's shape, an outline of thc character can be drawn and optionally filled in (figure llb). Thc curves used to draw thc characters arc cubic splincs which pass through thc points specified for thc letter (as contrasted with splines for which thc points control the shape of the curvc but do not lic on it, such as Bezier curves). For an example program, see figure 12.

METRFONT provides the designer with the ability to specify parameters which can then be used to modify a dcsign to produce different fonts in a particular family. The ability to spccify the thickness of thc pen allows a bold face to be

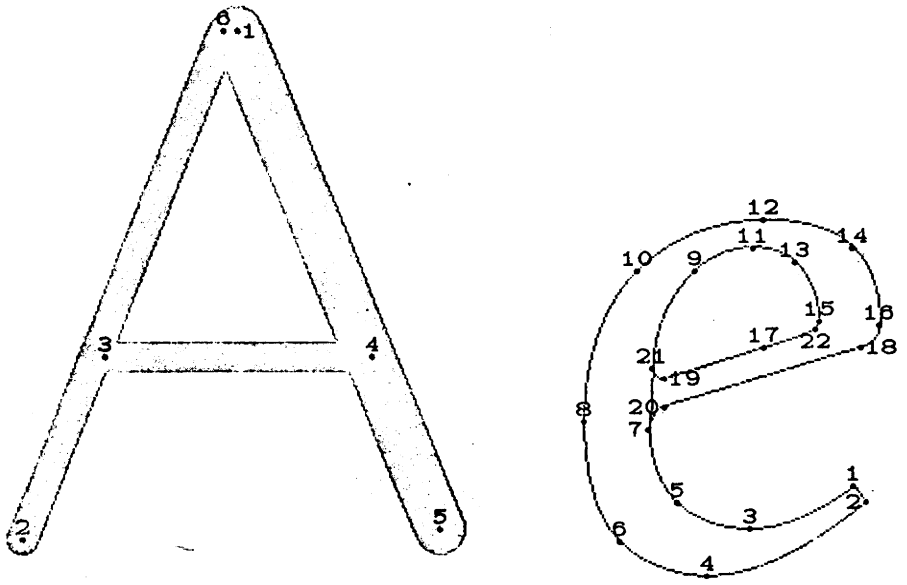

Figure JJ. a) Capital A drawn by METAFONT with a circular pen nib. b) Lower case e drawn as an outline.

created, and the ability to specify the slant of the image can produce an italic face. Other parameters that can be determined by the designer include those needed to change the serifs or the size and adjustments necessary to modify the face appropriately when the face is scaled very small or very large.

Several different output files are produced by METAFONT for use by various printers and terminal displays. One of the files contains descriptions of each of the characters in the font such as its width and height, for use by the formatting program TEX. Another file contains bitmaps of each character for use by a dot-matrix printer. Different resolutions of the output characters can be specified by the designer.

Although METAFONT can produce high quality output, it does have some drawbacks. To obtain. a high quality reproduction of an existing face, each character in a face must be converted into a METAFONT program, a process which can presently take several hours per character. Interaction with the system can be slow because the system is declarative rather than interactive, requiring the designer to first create the METAFONT program containing a mathematical specification of the letterforms, and then to run the METAFONT system with the letterform program. Modifications to a character are made to the program, and then the system is rerun. An additional drawback is that it is not possible to print out several characters to check on their spacing without using the TEX formatter.

```
% The letter A
%
%--------------------------------  --------------------------------
% specify where output goes
drawdisplay;                        % draw letter on screen
proofmode;                          % print proof sheet
%-------------------------------------------------------------------
charcode 'A;                        % this is a capital A
%-------------------------------------------------------------------
% position the points
x1=98;   y1=250;                    % x and y coordinates for each of
x2=0;    y2=0;                      % 6 points
x3=40;   y3=90; .
x4=170; y4=y3;
x5=105; y5=y1;
x6=203; y6=5;
%-------------------------------------------------------------------
% specify the pen
cpen;                               % circular pen nib
%-------------------------------------------------------------------
% draw the character                % using a pen width of 15, draw a line
15 draw 1..2;                       % between points 1 and 2, and between
   draw 3..4;                       % points 3 and 4; then with a pen
25 draw 5..6;                       % of width 25, draw a line between
                                    % points 5 and 6.
end
```

Figure 12. The **METAFONT** program for the capital A.

Letter Input Processor

The Letter Input Processor, or LetterIP as it is sometimes called, was developed by the Camex Corporation [6,21]. It is a modification of a system developed for the composition of display ads for newspapers and forms. With the addition of some software, its domain of application was extended to include the ability to create and store letterforms.

LetterIP is an interactive font digitizing system made up of three components: a character digitizer, an interactive character editing system, and a batch-processed data base for storage of the letterform information. It has a very high resolution vector display connected to a digitizer tablet with a menu and a puck. The process of entering and editing a character is done at the display terminal by a trained designer. Control points along the edge of the character are entered by pressing down on one of several buttons on top of the puck. The puck is also used to select one of the editing commands listed on a large menu to the left of the digitizer tablet. Letterforms are entered into the system by selectively entering one of three types of points, either a curve, tangent or edge point, from a photo or drawing

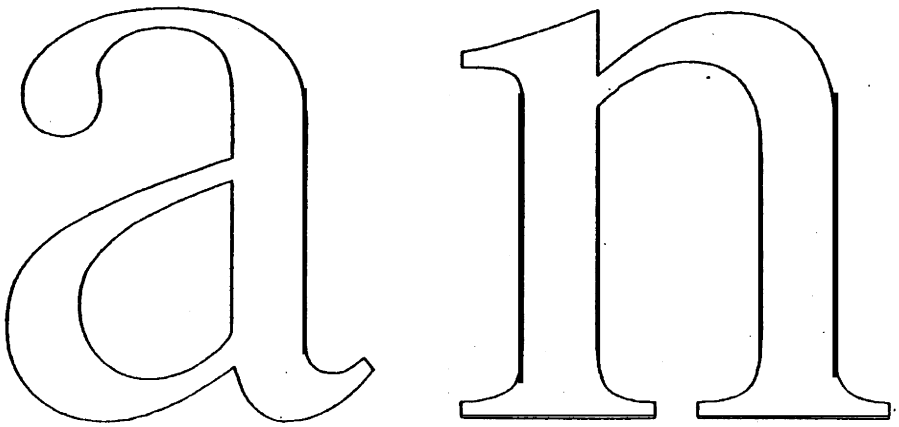

Figure 13. Outline letters produced by LetterIP.

of the letterform. As each point is entered, it is displayed simultaneously on the screen. Any gross errors or omissions are immediately detected and corrected. When all of the points have been entered, an outline of the character appears on the screen. The entered points are displayed at the designer's discretion. Feedback is thus available to the designer who wants to see the shape that has been input.

Once the letter has been entered, it can be edited interactively. Control points can be moved, added, or deleted. The ability to do *channelization*, or the lining up of points that lie within a specified channel, is provided. The outline of the character is updated with each change, but the previous outline can be retained on the screen for comparison if desired. The sidebearings, the left and right boundaries of a box enclosing the character, can also be entered and adjusted until they are positioned correctly. To aid in this process, additional letters such as N and O can be displayed to either side of the current character when the sidebearings are adjusted.

Several features are provided with LetterIP to aid in the editing process. A grid can be displayed along with each character to aid in the adjustment of the heights or widths of the images. A library is provided for the storage of character parts which can then be called up and used either as is, or they can be rotated, scaled or reflected before use. The ability to zoom and pan from half to double size is also available. Additionally, a gauge or ruler is provided for the measurement of stem widths, distance between points, or the angle between the ruler and the vertical axis, thus ensuring the accurate measurement of proportions of different characters.

The third part of the system, the data base, stores a model of the character outline as a line and arc model which may be output to a plotter either as an outline or a filled in shape (figure 13). A naming and directory structure is also provided as a means of storing the characters and accessing either a single character or a whole font. Output is also available on a magnetic tape.

The Camex' system was designed for the conversion of analog typefaces to a digital format for use by phototypesetters. The system does not produce bitmap images which can be used by dot matrix output devices. It also does not have the facility for displaying more than three characters on a screen for the designer to check on the appearance and spacing of the letterforms. Additionally, the system does not yet have an interpolation routine to produce different sized characters when given designs at either extreme, but addition of this routine is planned.

Other Design Systems

Several letterform design systems have been developed for some of the personal workstations developed in the last few years. The Xerox Alto has two varieties of programs which are used for letterform design [15]. The first, Fred, is a spline editor which provides features for fitting splines to a bitmap image that is displayed on the screen [1]. The bitmaps have to be input to the screen, though, before they can be edited, a process which requires additional equipment such as a scanner or camera to digitize the image. The second system, PrePress, is a font editor which allows a designer to define a letterform on the screen using a collection of cubic splines and straight lines connected by points [20]. This definition is converted by PrePress into raster images at a particular resolution and size for a variety of output devices.

The M.I.T. Lisp machine also has a spline based design system, called, appropriately, Font Edit [24]. This system does not differ significantly from PrePress, which may have served as the model on which Font Edit was based.

A variety of home-brewed bitmap editors have been written and are used to design letterforms. These systems usually require that the designer draw the letterform on graph paper, with the contour of the letter conforming to the resolution of the graph. This image is then entered square by square, with each square corresponding to one pixel. These systems have only limited popularity which is not surprising due to the large amount of time needed to enter each letter dot by dot. They are also not commercially available and remain local to the installation where they were developed.

Other work that is being done on problems associated with letterform designs include spline fitting algorithms developed by Michael Plass and Maureen Stone at the Xerox Palo Alto Research Center (PARC) [18]. 'Their algorithm finds a 'best fit' spline that conforms to the edges of a bit map. Additional work was done at Xerox PARC by John Warnock on producing grey scale images of typefaces [23].

Summary

There are several common aspects of each of the systems surveyed, and also many divergent trends. These similarities and differences are manifested in how

the letterform information is entered into the system, how the designs are output from the system, and the interaction of the designer with the system.

Each of the systems surveyed falls into one of two categories with respect to how the letterform type images are entered. The first category uses an encoding of a specific pattern of dots which then may be manipulated in one of two ways: the dot image may be edited using features of the system, or may be used to derive a descriptive image after which the digital image is discarded and only the description is saved. The second category of systems starts with a description of a character shape from which the dot pattern or bitmap can be derived. This description is entered through commands to the system and can be altered to produce changes in the letterforms. In either category, modification of the description allows new dot patterns to bc created. The final form of the stored data may be in the form of bitmaps or descriptions or both.

A design system may emulate the actions performed by a designer through tools or commands provided by the system which are similar to those used in a pencil and paper environment. The motion of a pen on paper is a common metaphor used in these systems, although the type of pen varies. METAFONT , for example, uses this approach to letterform design by providing an analogy to pen drawn letters. Movement of the pen is controlled by specifying the points through which the center of the pen will pass, thus producing ductal forms that look as though they were drawn using only strokes of a pen. The pen angle and the shape of the nib of the pen can be varied to produce variations in the characters. Other systems provide a glyptal approach, allowing points to be specified along the edges of a character so that outlines of the characters can be drawn and filled in. Use of an outline and a filler routine allows letters to be designed that could not be duplicated with only single strokes of a pen. METAFONT actually allows both types of drawing, though the pen metaphor is more powerful since there are more features which support it.

The audience to which these systems is addressed and the environment in which they are used affects both the quality of the letterforms generated and the success of the system in an environment outside of the original development testbed. Some of the systems, such as ITSLF and CSD, were only experimental and were never used by anyone other than the authors or anywhere other than where they were developed. Other systems, such as Ikarus, are marketed and licenced to companies which USC them for the production of digital designs. All of the systems, however, provide only input and editing capabilities; they are not systems that arc used for the actual design of letterforms. Pencil and paper techniques arc still used for the creation of new designs. Entering of data into these systems does not necessarily have to bc done by the designer but can bc done by persons trained to use the system and who have only a rudimentary knowledge of the letterforms. This lack of expertise can lead to degradation of the design when it is entered, as the subtleties of the design may be lost if they are not perceived or understood by the system operator.

The authors of letterform design systems have been mostly persons whose

background and experience has been with computers, not with type design. An exception to this is one of the authors of ELF, David Kindersley. His system provides features for the judgement of spacing of characters, a -problem which Kindersley has been concerned with for quite a while. In addition, Kindersley has worked on a companion system called Logos which provides a means of determining the optical center of a letter as an aid in determining the correct spacing. Although the author of Ikarus, Peter Karow, does not have a background in letterform design, his system provides several features to aid the letterform designer which are not supplied by the other systems such as interpolation to create a variety of designs given two extremes such as a light face and a bold one. This feature enables a designer to create only two designs and derive the remaining ones by a gradual metamorphosis of one of the designs into the other.

As a contrast to these systems, ITSLF and CSD, both developed in a computer environment, show an innovative approach to letterform design, but it is not one traditionally used by designers. As a result, both systems are interesting in their consideration of the problem, but not very successful in creating beautiful designs. METAFONT provides a more powerful tool through its provisions for the specification. of parameters such as the pen shapes, the slant of the characters, and its ability to create different sizes but it lacks any sort of visual interaction for specification of the design. The designer must manipulate programs which describe the letterforms and only after running the METAFONT system with these descriptions, is the actual letterform seen. The cycle of revision, creation, and examination of each character is a process which could more easily be done on an interactive basis with a pointing device and a video screen.

The conversion of a curved analog image into a digital one is not an easy process. The designer, when drawing a letterform, does not think about whether the curves are cubic splines, arcs of circles, or parts of a predefined curve such as the PM Spiral. Translation of a design into a description involves conversion of curves in the image into a representation of those curves so that they can be easily manipulated as the design is modified. The choice of which type of curve to use has not been resolved, although good results have been obtained with each of these three types of curves. There may exist yet another method which surpasses them, or perhaps a combination of the methods may produce better results. The type of curve chosen must satisfactorily reproduce the curves needed by the designer, but must not be allowed to control the design process.

Many of the systems attempt to capture the essential design of a typeface and then allow manipulation of the design through the use of parameters to control various features such as the size, weight, slant or the USC of serifs. This use of parameters within a design facilitates uniformity in creation of typefaces in allowing the designer to control the dimensions of the typeface independently of the letter designs themselves. These dimensions might include such things as the x-height or height of the lower case letters, the cap-height or height of the upper case letters, the height of the ascenders and depth of the descenders, the stem width, and the slant. If parameters are used to specify certain characteristics such as height or

slant, then a change in any of the parameters is applied to all of the characters which are controlled by that characteristic.

Defining the parameters that control a design is a difficult task. A type designer does not think about parameters when creating a new typeface, so it is difficult to quantize or measure the exact relationship between one letter and another. ITSLF allowed the designer to specify the values of certain parameters the letter E and then used those values to modify knowledge already stored in the system in order to create the remaining letters of the alphabet in a similar style. CSD stored information about characteristic parts of the alphabet so that the designer needed only to put the parts together to form letters. METAFONT provides the most flexibility of any of the systems, allowing the designer to specify both the parts and the letters and the relationships between them. One font created by the author of METAFONT had 28 parameters controlling the relationships between the letters [14]. Changing one of the parameters had the effect of changing one of the characteristics of the typeface. But is 28 the magic number? Can the essense of a design be captured in even a thousand parameters?

Conclusion

The use of letterforms is so widespread that most people do not even notice their existence, but instead take them for granted. This acceptance is indicative of the success of letterform design as images that are unreadable or unrecognizable do not gain widespread popularity. The subtletics and the sensitivities needed to create successful letterforms must be adequately translated into a new medium of expression or the medium will be useless for conveying information. With new devices capable of printing letterforms which are no longer analog but instead are digital, composed of hundreds or perhaps thousands of tiny dots each imperceptible to the eye, the creation of letterfonns must be studied and understood so that designs for this medium will succeed as well as the traditional forms. The use of the computer as a design tool can automate and extend the creation of these forms to fit the new medium, but an understanding of the artist and the art must preceed any attempts to capture and recreate the process. The letterform design systems surveyed here are each successful in identifying some of the problems, yet none of them has successfully reproduced the techniques of the designer, nor succeeded in integrating the artistic process with the automated one.

Acknowledgements---

This paper was written while I was a member of the digital typography research group at Stanford University whose support and encouragement is gratefully acknowledged. I would also like to thank Donald Knuth and Charles Bigelow for the loan of reference materials from their libraries, with additional thanks to Charles Bigelow and Andrew Cromarty for their careful reading and helpful criticisms of early drafts of the paper.

References

1. Baudetaire, P., *The Fred User's Manual*, Internal Report, Xerox Palo Alto Research Center, Palo Alto, California, 1976.

2. Берс,А.А., Представление, Преобразование Ипроектирование Н а рм Типографских ЩРифтов Для Фотонаьорных Абгоматов, Академия наук СССР Сибирское отделение Вычислительный Центр, Новосибирск, октябрь, 1977.

3. Bigclow, Charles, Technology and the Aesthetics of Type, Maintaining the Tradition in the Age of Electronics, *The Seybold Report,* 10(24):3-16, August 24, 1981.

4. Bigelow, Charles, The Principles of Digital Type, Quality Type for Low, Medium and High Resolution Printers, Part I, *The Seybold Report*, 11(11):3-23, February 8, 1982.

5. Bigelow, Charles, The Principles of Digital Type, Quality Type for Low, Medium and High Resolution Printers, Part II, *The Seybold Report*, 11(12): 10-19, 1982.

6. Camex Corporation, Letter Input Processor, Brochures from Camex, Boston, Ma., 1982.

7. Coueignoux, Philippe J.M., *Generation of Roman Printed Fonts,* PhD. Thesis, Massachusetts Institute of Technology, June 1975.

8. Hershey, Allen V., *Calligraphy for Computers,* Technical Report No. 2101, Computation and Analysis Laboratory, United States Naval Weapons Laboratory, Dahlgrcn, Virginia, August 1, 1967.

9. Hcrshcy, Allen V., A Computer System for Scientific Typography, *Cotnputer Graphics and Image Processing*, vol.4: 373-385, December, 1972.

10. Karow, Peter et al., Ikarus-System: computer-controlled font production for CRT and Lasercomp, Karow Rubow Wcbcr GmbH of Hamburg, Germany, September 1979.

11. Karow, Peter et al., Ikarus-System: computer-controlled font production for Photocomp, Karow Rubow Wcbcr GmbH of Hamburg, Germany, March 1979.

12. Kindcrslcy, David and Neil Wiseman, Computer Aided Letter Design, *Printing World,* October 31, 1979.

13. Knuth, Donald E., *Tex and Metafont, New Directions in Typesetting*, Digital Press and the American Mathematical Society, 1979.

14. Knuth, Donald E., The Concept of a Mcta-Font, *Visible Language* XVI(l), Winter 1982.

15. Lampson, Butler et *al., Alto User's Handbook,* Xerox Palo Alto Research Center, September 1979.

16. Mathews, M.V., Lochbaum, C., and Moss, J.A., Three-Fonts of Computer Drawn Letters, *Journal of Typographic Research* I(4), October 1967.

17. Merglcr, H.W., and Vargo, P.M., Computer Assisted Letter Design, *Journal of Typographic Research* II(4), October 1968.

18. Plass, Michael and Stone, Maureen, Curve-Fitting with Piecewise Parametric Cubics, Imaging Sciences Laboratory, Xerox Palo Alto Research Center, March 1983.

19. Purdy, Peter and McIntosh, Ronald, PM Digital Spiral, <Journal Unknown>, *Britain in Print, Forward Thinking,* 1978.

20. Ramshaw, Lyle and LaPrade, Kerry A., Prepress Manual, version 2.1, Xerox Corporation, September 1980.

21. Richmond, Wendy, Letter Input Processor, Internal Memo, Camex Corporation, October 7, 1982.

22. Seybold, Jonathan, Digitized Type: What is it? What does it mean for typesetting and word processing?, *The Seybold Report,* 24:3-17, August 27, 1979.

23. Wamock, John, The Display of Characters Using Gray Level Sample Arrays, STGGRAPH 80 Conference Proceedings, Seattle, Washington, July 14-18, 1980, pp.302-307.

24. Weinrab, D. and Moon, D., *Lisp Machine Manual,* 1981, Massachusetts Institute of Technology.

25. Wiscman, N.E., Use *ELI: for your Elfabets,* University of Cambridge Computer Laboratory, Rainbow Memo 162, July 18, 1978.