Programming in PostScript

Techniques and applications in vision research, plus a survey of PostScript hardware and software

Denis G. Pelli

Most people who use PostScript-driven printers have never seen a PostScript program because their application software generates and sends the necessary PostScript commands automatically whenever they say “Print.” For routine applications, this is as it should be. However, programming directly in PostScript opens up many new possibilities, particularly for creating images from mathematical descriptions.

For instance, I use PostScript to produce patterns to help my study of the visual system. In this article, I will use some of those patterns to illustrate PostScript’s power. I’ll also list some useful PostScript products and services.

PostScript programming is best for those occasions where you begin by knowing only a mathematical description of an image and want to find out what the image would look like. This approach is old hat to anyone who has done computer graphics on a video display. Unfortunately, it is difficult and expensive to obtain an accurate photograph of what’s on the screen, primarily because of the non-linear intensity and variability of photography. PostScript lets you get faithful renditions directly on paper.

Another attraction of PostScript is that you can debug and refine your picture-generating programs using a relatively low-cost (around $5000) laser printer such as the Apple LaserWriter, which will produce fair-quality drafts (300 dots per inch) immediately for a few cents per page. For the final output, you can send the same program by modem to a $60,000 printer like the Linotype L300, which will produce superb quality prints (2540 dpi) for $10 per page. The same program runs on both printers because PostScript is machine independent. (This article includes examples produced by both printers.)

Overview

PostScript is a full-fledged structured language. Like FORTH, it is stack oriented and uses postfix notation. For example, to add 2 and 2 you say 2 2 add. Each PostScript printer has a built-in microprocessor and a PostScript interpreter that will run your program. The page is represented by a bit map, with 1 bit per pixel. (PostScript incorporates some limited support for color, but none of the available printers supports color.) When the page is printed (usually by the showpage command), the bit map is transferred to the page, making each pixel either black or white. However, to preserve machine independence, you are not allowed to read from or write to the bit map. Instead, your program includes “painting” operators that set and clear bits within defined areas of the bit map.

PostScript offers three kinds of painting operators, allowing three quite different approaches to describing your image: one-dimensional paths, two-dimensional sampled images, and text. Before explaining the three kinds of painting, I should say a few words about an operation that can be applied to all three: digital halftoning.

Each pixel in a PostScript image is either black or white. PostScript produces the illusion of intermediate gray levels by digital halftoning—breaking the image up into cells, each containing many pixels, and setting some of the pixels to white and the rest to black. If the cells are too small for the eye to resolve, the viewer will see a uniform gray, and the illusion will be successful.

You can adjust the size of the cell by the screen operator, which lets you specify how many cells per inch. When you request a particular gray level, PostScript will set approximately the fraction of the pixels in each cell to white and the rest to black.

Obviously, there’s a trade-off here. You’d like to have very small cells so that the eye can’t resolve them and the picture will look continuous. But if the cells are small, they will contain few pixels and you will have only a few possible gray levels.

Your gray levels will be coarsely quantized (approximated by the nearest available discrete level). In images where the gray level varies smoothly, such as in a person’s face, coarsely quantizing the gray level will produce noticeable contours at the transition between gray levels.

Note that digital halftoning is quite new and the terminology has not yet settled. In particular, the word “dot” refers continued

Denis G. Pelli is a visual psychophysicist studying the information rate of the human visual system. He is an associate professor at the Institute for Sensory Research, Syracuse University, Syracuse, NY 13244-5290.

May 1987 • Byte 185
to two quite different things. If the resolution of a printer is specified as "300 dpi," it means 300 pixels per inch. If a halftone image is "18.75 dpi," it means 18.75 halftone cells per inch. If the image is produced by digital halftoning, you can assume that each cell has many pixels. For example, nearly all the black-and-white continuous-tone images in BYTE have 120 cells per inch. Most of those images were produced by traditional halftoning, which does not involve pixels at all.

The faces below (see figure 2) were produced by digital halftoning with 120 cells per inch and 2540 pixels per inch, so each square cell contains (2540/120)^2 = 448 pixels. In principle, this would allow for 449 different gray levels (from 0 to 448 pixels set to white in each cell), but an undocumented limitation of PostScript is that the requested gray level is quantized to 8 bits, allowing at the most, 256 different gray levels.

Paths

One of the fundamental design goals of PostScript is to be independent of the details—especially the dot resolution—of the particular printer used. To achieve and enforce this, PostScript does not let you ask any questions about the state (black or white) of the pixels that make up the page’s image. Instead, you create an image by defining an area or “path” and then filling it.

A path is a series of lines and curves, like a line drawing. However, the path itself is not a drawing. To make an image, you must use a painting operator. The two painting operators that can be applied to paths are stroke and fill. Stroking a path produces an image on the page consisting of a line of specified thickness. (You can also specify the gray level, the kind of dashes for a dashed line, the type of line termination, and the shape of corners.) Filling a path causes areas that are inside the path to be painted with a specified gray level, without affecting areas that are outside. The meanings of “inside” and “outside” are obvious for a simple closed curve. For more complex paths, PostScript offers a choice of two different rules for determining which areas are inside and out.

Figure 1 was produced by stroking paths. The R and V are two of the ten characters (C D H K N O R S V Z) in the Sloan font, named after its designer, Louise Sloan. The Sloan font is recommended for vision testing (reference 1).

Obviously, if you are designing vision tests, it is helpful to be able to use the officially sanctioned font. Listing 1 shows the programs that I wrote to produce R and V. I wrote similar programs for the other eight characters in the Sloan alphabet. With some extra statements, you could make this a proper PostScript font, usable in any place where you would like to display text.

The first two statements in listing 1 are comments. Everything from a % to the end of the line is ignored. It is traditional for the first line of a PostScript program to begin with %! to help your computer identify it as a PostScript file. The third line begins /preSloan, which pushes the name preSloan onto the stack. Then comes a procedure enclosed in braces that extends over several lines. PostScript treats spaces and carriage returns merely as separators. The procedure is compiled and pushed onto the stack as one item. After the closing brace of the procedure is the word def. That causes PostScript to pop the top two items from the stack and use these to define the word preSloan to refer to the given procedure. It does that by putting the word preSloan in a dictionary, with the procedure as the meaning. From now on, the word preSloan will invoke that procedure.

The following lines similarly define the procedures postSloan, R, and V. The line 50 700 moveto first pushes 50 onto the stack, then pushes 700 onto the stack. Then the operator moveto pops the top two items from the stack and moves the current location to that x-y coordinate. (The default coordinate system puts the origin at the lower left corner of the page and measures distance in “points,” 72 to the inch, so the location [50,700] corresponds to (0.69 inches, 9.72 inches), which is 0.69 inches from the left margin and 11 - 9.72 = 1.28 inches down from the top of the 8½- by 11-inch page.)

The 40 40 scale command amplifies the scale so that one unit corresponds to 40 points. R invokes the R procedure, which draws an R one unit wide and one unit high (where one unit is now 40 points). The R is made up of lines and arcs. It is all “clipped” by a unit square, set up in preSloan, so that only what falls inside the square appears in the final image. It is the same for V. Finally, showpage transfers the image to paper and ejects the page from the printer. You can send this program to any PostScript printer to obtain an image like figure 1.

Figure 2 was also produced by paths.

---

**Figure 1:** Two letters of the Sloan font for vision testing produced by listing 1 on a Linotype L300 at 2540 dots per inch.

**Figure 2:** Faces for testing infant acuity produced by listing 2 on a Linotype L300 at 2540 dots per inch.
The face in the figure is designed for acuity testing of infants. When you can see anything at all, you can see that it’s a face. The black and white lines making up the face disappear when they are too thin for the eye to resolve. The biggest face will disappear when you are about 24 feet away, depending on your visual acuity. The smaller faces will disappear at proportionally shorter distances. The smallest face will disappear at about 18 inches. The trick to making it disappear is to draw it with a ribbon consisting of equal parts of white and black, so that when the white and black are blended together by the eye, the resulting average is identical to the gray background (references 2 and 3).

Listing 2 gives the PostScript program used to produce figure 2. After a few comments the face procedure is defined; it consists solely of moveto and lineto commands that trace a path. The draw procedure is defined next. Then the whole page is painted gray.

The program is optimized for the Linotype L300 running at 2540 dpi. For best results at 300 dpi, as such as on a LaserWriter, you should remove the 0% from the beginnings of two lines labeled “at 300 per” and place the 0% symbol at the beginning of the two lines marked “at 2540 dots per inch.” The 0% causes the rest of the line to be treated as a comment—in other words, ignored. After you remove the 0% from each line, those lines will give you the best choice of screen density for a LaserWriter.

Note that this does not contradict the claim that PostScript is device independent. When you go from higher resolution to lower resolution, you have to make some trade-offs in coarseness of screen and precision of gray level. You can often get acceptable results by letting the PostScript printer make the trade-off itself, but if you understand the underlying issues, you can usually get better results by specifying exactly what you want. The two gray levels were found by trial and error to produce 50 percent reflectance on the two printers at these screen densities.

The main program is next. It consists of a loop 91... repeat, which means that the procedure enclosed in braces should be repeated nine times. Inside the braces you will find the commands face, which lays down the path of the face, and draw, which paints the path, and the operators translate and scale, which shift and shrink the coordinate system to get ready for the next face.

To produce a black-and-white ribbon, the draw procedure must stroke the same path twice. This requires a trick, since
strocking a path eliminates the path. The first command in draw is gsave, which saves the graphics state, including the path. Then the line width is set, the gray level is set to 0 (i.e., black), and the path is stroked. This produces a black line drawing of a face on the gray background. Next, the grestore command restores the graphics state, including the path but not affecting the image, which is not part of the graphics state. Now, after reducing the line width by half and setting the gray level to 1 (i.e., white) draw strokes the path again. This produces a white line running down the center of every black line. After the nine iterations of the main loop, the image is done. All that remains to put it on paper is to give the showpage command.

A reasonable question to ask at this point is: How did I obtain the sequence of lineto and moveto commands to produce the face? I could have drawn the face in pencil on graph paper and read off the coordinates. In fact, I obtained this sequence by using MacDraw to draw the face frehand. I then captured the PostScript output of MacDraw and cleaned that up using Post-A-Matic (a program that translates Macintosh's abbreviated PostScript to standard PostScript commands) and a bit of editing. A new program called Cricket Draw makes this much easier. (Further information on items mentioned appears in the text box at left.)

To test an infant's vision, I would print up many gray cards, each with a single face on either the left or the right. The testing technique is called "Forced Choice Preferential Looking." A person who can't see the test card looks at the infant's eyes and tries to guess on which side the face is by which way the infant is looking. By testing many times, you can determine how reliably the person observing the infant's eyes can guess where the face is. If the infant's eyes tell you where the face is, you can reasonably conclude that the infant can see the face (reference 4).

Sampled Images
The second way to produce an image in PostScript is to use the image operator and provide it with the gray level of the desired image at each point on a regular grid. The most obvious application of this would be to reproduce a photograph that has been sampled previously. Figure 3a is an example. Each sample from the photograph is imaged by a cell containing many pixels. The fraction of the pixels in the cell that are white determines the gray level. Figure 3a was produced by a Linotronic L300 at 2540 pixels continually.
IBM just announced their new redesigned "standard" keyboard for personal computers. There's only one problem: it won't work on your IBM computer if it was purchased prior to June 1986. IBM has caught up to users who believe there is a choice.

Not to worry. Our new Turbo-101 Enhanced Keyboard gives you the layout and enhancements of the IBM with some logical improvements (see above). And it works on your existing PC, XT, AT, PCjr, AT&T, Epson and virtually all compatibles!

The Turbo-101 is the best data entry tool since the pencil! For users of spreadsheets like 1-2-3, the Turbo-101's separate cursor controls and numeric keypad makes entering numeric data into cells and moving from cell to cell as natural as moving your fingers. And for word-processing, the 'selectric' typewriter layout makes the Turbo-101 as easy to use as a pencil, and with the extra large Enter, Shift & Control Keys, you'll make so few mistakes, you won't even need an eraser!

SPECIAL OFFER!
ONLY $149.95 FOR BOTH KEYBOARD & SOFTWARE
Includes 30-day money back guarantee and 2 year full warranty. To prove that we don't ignore you or your pocketbook, you get our Turbo-101 Enhanced Keyboard and Borland's Turbo Lightning for an astounding $149.95.* No, you didn't read it wrong. During this amazing introductory offer you get both keyboard and software for less than most software programs by themselves! Now, if you're still feeling ignored, you can always do what you know-who wants you to do... and buy a new computer to get their keyboard!

Up to now, DataDesk International may be one of the best kept secrets, but here's what's being said about our first end user Keyboard/Borland software bundle:

"Who Can Pass Up a Deal?"
Department Talk about an aggressive product!
John C. Davis
InfoWorld May '86
It's a great keyboard!
"It takes all of the problems exhibited by their regular PC XT keyboard... It's a great bargain!"
P. C. Productivity Digest Aug '86
"Borland's latest, best designed portable keyboard has reduced response... It's a hardware bargain of the year!"...says...says
Barbara Pounder, Oregonian Jun '86
"It's a good keyboard. Good feel, the keys have tactile feedback. No multi-key contacts. This is about as good a keyboard deal as you're likely to find... I have absolutely no hesitation in recommending the Model PCX-2000."
Jerry Nowell, Byte Magazine Aug '86
"The keyboard is neat to type on and feels solid. It has tactile feedback keys. I can type much faster on it." Test Drive Scorecard: DataDesk 20 Key Entry & Telephonics Magazine May '86

"The best part of the keyboard is the way it feels. It's ideal and fast. I've never worked on a keyboard with a better feel." Business Computer Digest Aug '86

credit cards orders call
(800) 826-5398
in CA call
(800) 592-9606

BLACK TURBO-101 ENHANCED KEYBOARD AND BORLAND'S TURBO LIGHTNING SOFTWARE FOR ONLY:
$149.95 LIMITED OFFER
NO UNIT PRICE! VISA CHECK
AMOUNT ENCLOSED:

MADE IN USA

The Only Alternative
datadesk INTERNATIONAL
7550 Haskell Avenue
Van Nuys, California 91406 (818) 780-1673
Turbo-101 is trademark of DataDesk International. Turbo-Lightning is a trademark of Borland International. IBM and AT are registered trademarks of International Business Machines Inc. Inquiry #88 for End Users. Inquiry #89 for DEALERS ONLY.
per inch. Figure 3b was produced by sending exactly the same program to an Apple LaserWriter Plus, which prints at 300 pixels per inch.

However, there is a more interesting application of the PostScript image command. You can use mathematics to synthesize an image that never existed before. Figure 4a is an example. By just looking at this pattern, you can perform a systems analysis of your visual system.

Any patch of the pattern is a sinusoidal grating, varying from white to black and back again. However, from left to right across the image, the fineness of the grating increases until it disappears. Similarly, from bottom to top the contrast of the grating decreases until it disappears. The lower the contrast of the grating, the higher the contrast sensitivity required to see it. If you follow any one bar in the image from the bottom up until it disappears, the height at which it disappears is the contrast sensitivity of your visual system for a grating of that fineness.

The fineness of a grating is usually specified as a spatial frequency—in cycles per inch, for example, where a cycle is a pair of light and dark stripes. However, the sensitivity of your eye depends not on the number of cycles per inch but on the number of cycles per degree of visual angle subtended at the eye (try looking at the pattern from various distances), so for vision work it is preferable to specify spatial frequency in cycles per degree.

The outline of visibility of all the bars describes the sensitivity of your visual system over a wide range of spatial frequencies. Notice that the vertical bars when taken together form a hump, peaking in the middle and falling off toward the left and right. The falloff to the right is determined largely by the optical quality of your eye. The falloff to the left is determined by neural limitations of your eye. This outline of visibility explains many characteristics of vision, including many visual illusions.

Figure 4a was produced by the program in listing 3. After a few commands to set up the coordinate systems of the sampled image and the page, the equation for the image is supplied to the image operator. The equation is essentially

\[ g = 0.5 \left(1 + 10^{-2} \sin 10^5 \right), \]

where \( g \) is the desired gray level (i.e., re-

### Figure 3: Digitized image of the San Francisco skyline. Program and image file courtesy of Adobe Systems Inc. (Included on the COLOPHON 2 IMAGES disk; see the text box.) 3a: Produced on a Linotype L300 at 2540 dots per inch. 3b: Produced on a LaserWriter Plus at 300 dpi.
Until now, working away from your office meant having to settle for lesser computers—or no computer at all.

Toshiba ends that compromise. The T3100 delivers the power of an IBM® PC AT™ in a lean, 15-pound computer that you can take with you. With a full-sized keyboard and advanced gas plasma display that give you equal comfort and clarity on your desk—or on the road.

Don't let its size fool you.

Though the T3100 is no bigger than many laptop computers, it offers the power of a full-function IBM compatible desktop AT system. You get the speed and ease of a built-in hard disk, communications and expansion options, and more—including dual-voltage power that lets you plug in and go to work just about anywhere in the world.

*Okay, we exaggerated. It's actually 14.2" x 12.2". But you really get a 10.6MB built-in hard disk.

See the system that has all of the industry experts so excited. Call: 1-800-457-7777 for your nearest Toshiba computer and printer dealer.

In Touch with Tomorrow
TOSHIBA
TOSHIBA AMERICA, INC., Information Systems Division

flectance), and \(x\) and \(y\) specify the position on the page. As \(x\) increases, the argument of the sine function, namely \(10^x\), increases more and more rapidly and the value of sine \(10^x\) oscillates between \(-1\) and \(+1\) more and more rapidly. The program figures out the resolution of the printer and chooses the maximum spatial frequency (and thus the maximum value of \(x\)) accordingly. This is because proper representation of a sinusoid requires at least two samples per cycle. If the grating is so fine that fewer than two samples occur in each cycle (a pair of light and dark stripes), there will be a moiré effect, resulting in very broad stripes where there should be fine stripes.

The PostScript printers are not smart enough to take into account the content of your image in their efforts to print the image faithfully at whatever the printer's resolution happens to be. Thus, programmers must watch out for moiré effects. Figure 4a was produced by running listing 3 on a Linotype L300 at 2540 dpi and 127 cells per inch. Figure 4b was produced by running the same program on an Apple LaserWriter Plus at 300 dpi and 18.75 cells per inch. Note that figures 4a and 4b represent different images because the program explicitly changed the image to best take advantage of each printer's resolution and avoid moiré effects.

This kind of image, a sinusoidal grating swept in frequency and contrast, was first produced by Fergus Campbell and John Robson at Cambridge University in the early sixties (reference 5). John Robson explained to me how they did it. They used two sawtooth oscillators to sweep the beam of an \(x-y\) oscilloscope in a vertical raster pattern, so that one vertical line took about 20 milliseconds and a whole frame took about a minute. The sinusoidal grating was produced by a sinusoidal oscillator connected (via a relay and capacitor) to the oscilloscope's intensity-control input.

Campbell and Robson produced the vertical fading in contrast by using a high-speed relay to quickly charge a capacitor to the oscillator voltage at the bottom of each vertical line and letting the capacitor discharge through a resistor as the beam swept upward. The horizontal variation in frequency (fineness of the grating) was produced by manually adjusting the frequency of a sinusoidal oscillator.

Figure 4a: Sinusoidal grating swept in frequency and contrast. Produced by listing 3 on a Linotype L300 at 2540 dots per inch and 127 cells per inch.

continued
FEATURISTIC PROTECTION
The PC1025 by DATA SPEC

With full-featured convenience and sleek contemporary styling, the PC1025 is a new dimension in control protection.
Its slim-line design, color-matched to your PC, fits easily on top of your computer without intruding into your workspace.

The PC1025 combines the complete safety of system and modem surge protection, the easy convenience of five lighted power switches and a Master Switch with the versatility of a built-in data switch. Added features include a digital clock, LED status lights, and more.

FEATURISTIC protection that gives you more system control and convenience than you thought possible from one source.
CP/M Emulation

Run CP/M software on your IBM-compatible personal computer. We have Z80-HD64180 co-processor cards for 280 or software emulation. For $99.95 our Blue Thunder Super-speed crunches through your work. Almost as fast is the 10-MHz High Speed model at $395.95. The 6-MHz Standard model is $249.95. Including software.

Blue Thunder $249.95

HD64180

Hitachi’s new microprocessor has taken the Z80 world by storm. Z80-compatable, but 15% faster for the same clock speed. Counter-timers, serial ports and DMA channels on chip. CMOS. Our customers have reported instances to us where the HD64180 blows away an 80186 at similar clock speeds. This baby moves!

Our $295.00 D64180LP co-processor comes with our CP/M emulators and real-time programming support. 2-RS232 serial ports and 64K of memory. An excellent basis for a Z80-HD64180 development system. Clock speeds to 8.215 MHZ available.

D64180LP $295.00

For $445.00 you can get our high powered D64180 real-time co-processor. 256K memory, expandable to 512K 2-SBK bus connectors which allow you to use hundreds of commercially available interface modules, such as synchronous communications, disk interface, etc.

Software Developers

Let us configure a complete Z80 or HD64180 development system for your PC. Our Remote Debugger eliminates the need for an ICE. We have VAX and PDP-11 co-processors too.

ZWorld

2065 Martin Ave. #110
Santa Clara, CA 95050
(408) 980-1678

In Germany: System Tel 0831/1687

POSTSCRIPT

Listing 3: The PostScript program to produce the sinusoidal grating test in figure 4a.

%! Ps Produce Figure 4
/inch {72 mul} def
/width 4.54 inch def
50 400 translate
% make square image of desired width
width width scale
/dpi
72 0 matrix defaultmatrix dtransform dup mul exch dup mul add sqrt def
/screen dpi 16 div def
/screen 127 gt [/screen 127 def] if
/screen currentscreen 3 -1 roll pop setscreen
% number of cells across image
/n width screen mul 72 div 2 sqrt mul 0.99 add cvi def
/ny n 2 mul def
/ny n 2 idiv def
/tbase 0.5 n mul 1.0 nx div exp def
/Ia 360 nx div tbase ln div def
/c 1 def
/cbase 0.003 1.0 ny div exp def
/S nx string def

nx ny 8 [nx 0 0 ny 0 0]
|/c c cbase mul def /f f def
0 1 nx 1 sub[S exch f sin c mul 1.0 0 0 126.5 mul cvi
put /f f tbase mul def]for
S| image

showpage

cillator as the beam swept across the screen.

Campbell read out the desired frequencies from a table, and Robson adjusted the dial. They recorded the image on film by taking a long-exposure photograph.

Working with Text

The third way to produce an image in PostScript is to use text.

PostScript makes it very easy to create characters of arbitrary size and rotation. In fact, the font machinery for producing text is built from the commands for producing paths. Each character is created by its own program. Typically, a program to produce a character creates a path and then fills or strokes it. However, because writing text is such an important use of PostScript, several things were done to make it easy and fast.

First, fonts are stored in special dictionaries inside the printer, with a little program for each character, and special commands are provided for specifying what font you want to use and at what size. Commands are also provided to produce a whole line of text from an ASCII string, which is just a list of the desired characters. There are even facilities for kerning, or adjusting, the spacing between particular letter pairs, such as "TA," which looks best if the "A" is tucked slightly under the hat of the "T."

Second, it takes a lot of computation to run the programs correspondingly to each character in a page of text. To alleviate this burden, whenever PostScript computes a character to paint it onto the page, it also saves a copy of the character's bit map in a special area of memory called the "font cache." If the character is needed again, PostScript retrieves it from the cache instead of computing it again. Naturally, the cached copy can be used only for the same character from the same font at the same size and rotation.

Figure 5 (produced by listing 4) is a simple example of the use of text. This is a test chart to measure visual acuity, much like one you might see in an optometrist's office. The letters are all at high contrast and get smaller and smaller until they disappear. The smallest letter that you can read is a measure of the resolution limit of your eye.

continued
Aztec C ... The Best C Frees the genius in you

You've got a great idea...you're ready to write your programs.

You don't want to be sidetracked by all the paperwork. With Manx Aztec C and the ingenious make function, your creative processes won't get bogged down in program administration and housekeeping. Manx Aztec C has the most sophisticated,hardworking program administrator available to you. Once you've described your project, adding new features or enhancements is simple. You never have to concern yourself with the repetitive, tedious task of rebuilding your systems.

The development process moves quickly. Compiles, assemblies, link edits...all finish in record time.

Manx Aztec C is the fastest, most efficient C development system in the industry. Benchmarks show it...reviews commend it...users praise it.

You're ready to test the program. You're ahead of schedule. The Manx Aztec C Source Level Debugger shows you the exact C language statement giving you a problem. You fix the problem quickly...you're still ahead of schedule.

You've got some time for fine tuning. The Manx Aztec C Profiler examines your program, tells you where the slow spots are and validates your test procedure. A few changes and it's exactly what you wanted.

You've made it!

Aztec C is available for MS-DOS/PC DOS. Call for details on Macintosh, Amiga, Apple II, CP/M-80, CP/M-86, TRS-80, ROM and others.

To order, or for information

Call Today
1-800-221-0440

In NJ or outside the USA call (201) 542-2121

30-day satisfaction guarantee. Special Discounts are available to professors, students, independent developers, and on a "trade-in" basis. Site licenses.
Figure 6 (produced by listing 5) is a new kind of test chart. The letters are all the same size, but they vary in contrast, fading out until they disappear. The lowest contrast that you can read is a measure of the contrast sensitivity of your eye. The combination of acuity and contrast sensitivity results in a more complete assessment of the visual function of your eye than would acuity alone.

Incidentally, you may note a similarity between figure 4a and figures 5 and 6. In fact, the finest letter that you can read provides a useful estimate of the finest grating that you can resolve (the right edge of the hump in figure 4a), and the lowest contrast letter that you can read provides a useful estimate of your contrast sensitivity for gratings—the height of the hump in figure 4a (reference 6). Figures 5 and 6 were produced by moving to a new location on the page for each character, setting the gray level, setting the font size, and printing the character by the SHOW command. The programs are given in listings 4 and 5.

Note that figures 5 and 6 were produced with the Sloan font, which you won’t have. The programs in listings 4 and 5 explicitly check to see if Sloan is available; if not, they substitute Helvetica.

Tools for Programming in PostScript
First you’ll need the PostScript Language Reference Manual and the PostScript Language Tutorial and Cookbook, both written by Adobe Systems, the company that invented PostScript and put it in the public domain. These clear, well-organized books are models for how manuals should be written, with a clear separation of the tutorial and reference functions. The tutorial has lots of real examples, and the reference manual is organized alphabetically.

You’ll also need a computer with a plain text editor (to write your program) and connection to a PostScript printer. Your computer won’t be doing much, just sending your program to the printer and receiving any messages from the printer. It’s the printer that interprets and runs your program.

The simplest way to connect any computer to a PostScript printer is by a serial line, using any simple communication program. However, most users do not

continued
Our Development Tools Generate A Lot of Positive Feedback.

Jeffrey Zarkow, President of Avocet Systems

Microprocessor development tools without the shock of a big price tag.
The AVMAC family of assemblers from Avocet lets you develop microprocessor code on your personal computer or VAX. Avocet has been developing PC software since before there were PC's. Our products have been refined over several generations giving you powerful and flexible development tools. They work with your existing operating system and text editor, so they're easy to install and easy to use.

Do all of your development on a single PC or VAX without switching systems.
If your next project calls for a different microprocessor, just call Avocet. There are Avocet assemblers for most common microprocessor families. So modifying your development system is as easy as changing diskettes. Check the chart—the assembler you need is probably there.

With Avocet, you're wired into a family of support products you can trust.
Each assembler package includes the AVLINK linker, AVLIB librarian, AVREF cross-reference generator and other utilities. And Avocet also offers you a host of other tools you need, such as TRICE self-powered in-circuit emulators, AVPROM programmers, text editors and the AVSIM full-screen simulator/debugger. So the PC on your desk can hold a workbench full of development tools.

Try before you buy. And we'll ship in 48 hours or less.
You probably need your tools right away. And that's when you get them. When you order from Avocet, we'll also send you a complete demonstration kit for both our assembler and our AVSIM simulator/debugger.
Try the demo for thirty days—if you aren't completely satisfied, we'll refund your purchase price, less $35.00 for the demo disk. That's yours to keep, which is a good deal by itself! We're sure you'll find Avocet the best connection you've ever made.

Call 1-800-448-8500.
Avocet Systems Inc., 120 Union St.
P.O. Box 490, Rockport, ME 04856
Inquiries 34

AVOCET SYSTEMS, INC.
own a PostScript printer. Typically, they share one over a network. This arrangement requires that you use a downloader program to send your PostScript program (a text file) via the network to the printer. A good downloader should also be able to receive, display, and save any messages sent back from the PostScript printer.

Receiving messages from the printer is very important; otherwise, you'll only know when your program crashes by the fact that no paper comes out of the printer. I debugged my first few PostScript programs that way, running to the other room, waiting to see if I got any output—usually I didn't—and walking back to edit my program a bit and trying again. It's reminiscent of using the toggles on a PDP-11/20 to program in machine code—obviously something best left to "real programmers," as no sane person would enjoy such a tedious procedure.

If you can't receive messages from the printer, a less convenient solution is to use Adobe's free "error break page handler." A PostScript program you send to the printer once. From then on (until you reset the printer), when any program crashes you get a printed page showing the error message, the offending command, and a peek at what's on the stack. However, this still requires running back and forth between your computer and the printer.

Several good downloader programs are now available for the Macintosh. The best one is free. SendPS from Adobe Systems will send your program over AppleTalk to the Apple LaserWriter. It will also receive any messages sent back over AppleTalk from the printer, show them to you as they arrive, and automatically store them in a file for your later perusal. You can also ask for a report on the status of the printer at any time, and you can initiate a remote reset of the printer. SendPS doesn't include a text editor, so you'll need one. (I recommend the shareware desk accessory MockWrite.) PostHaste and JustText have all the features of SendPS (except showing messages as they arrive), plus built-in multindow editing, and JustText comes with extra programs that convert MacPaint, Thunderscan, and MacVision files into PostScript.

Cricket Draw, from Cricket Software, lets you edit and download a PostScript program but does not receive messages from the printer. Its main attraction for the PostScript programmer is that it lets you make a MacDraw-like drawing (or read in a MacDraw file) and convert it into pure PostScript. It has a handy PostScript help function that lets you look up a brief explanation of any PostScript operator—moveTo, for example.

One feature I'd like to see in future PostScript downloaders is an include filename statement that would indicate to the downloader that another file (e.g., containing your favorite subroutines) should be included when the program is downloaded. This is a standard feature of most modern languages, but PostScript lacks it and thus deprives the programmer of many of the main advantages of modular programming. You have to put everything in one file, making nontrivial PostScript source files unreasonably long and making it difficult to keep all your copies of a given subroutine up to date.

At present, much less software is available for PostScript downloading for non-Macintosh computers. Synergetics sells some utilities for programming in PostScript on an Apple IIGS, IIC, or IIE. Several companies offer hardware and software (see table 1) to connect an IBM PC-compatible to an AppleTalk network; some of these products let you receive messages from the printer. Now that IBM has endorsed PostScript for use in its future desktop publishing products, things are bound to improve for IBM PC users.

Many computer companies, such as Digital Equipment Corporation and Sun Microsystems, are bringing out new PostScript printers and providing them with interfaces to connect to computers from those manufacturers. (A list of PostScript printers appears in the text box on page 188.) Most of the existing and forthcoming printers have 300-dpi resolution, like the Apple LaserWriter.

Linotype makes two high-resolution

---

Figure 5: Conventional acuity chart produced by listing 4 on a Linotype L300 at 2540 dots per inch.

Figure 6: Pelli-Robson Letter Sensitivity Chart produced by listing 5 on a Linotype L300 at 2540 dots per inch. Copyright 1987 by Denis G. Pelli.
We've taken the work out of doing Windows.

Microsoft® Windows is becoming the most popular operating environment for PC systems. It's not surprising. Windows provides the foundation for an exciting new generation of applications that users are demanding. In addition, Windows handles many of the details involved in a software project allowing you to spend more time enhancing your application. That's why a growing number of corporate and independent software developers are building Windows applications.

The Microsoft Windows Software Development Kit is your key to this extraordinary new environment. It's packed with full reference documentation, libraries, utilities and sample programs. Together with our C Compiler or Macro Assembler, it's a comprehensive package that lets you make the most of your application.

Software with a new view.

Giving your applications the Windows treatment begins with a new look. The rich graphical environment allows you to rethink how you want your program to be presented on screen. It lets you mix text and graphics. You can incorporate multiple fonts in a variety of sizes, faces and styles. And it provides the basic building blocks that make it easy to create drop-down menus, dialog boxes, scroll bars, icons and more.

These features not only simplify your application design, but also provide the familiar interface that makes your software easier to learn and use.

Easing the data shuffle.

Of course, there's more to Windows than just looks. Now, different applications can work together. In concert.

The Windows Clipboard provides support for users to cut and paste information between your applications and others. Or you can use messages to establish "hot links" to transfer data automatically.

Upgrading made easy.

Windows' device independent design allows you to build your application today and take advantage of new technology as it becomes available. When new graphics cards, printers and pointing devices appear they can be used with your software, without modifying your code. Simply by installing the new driver.

Your window of opportunity.

The Microsoft Windows Software Development Kit is your fastest route to better applications. And with it, we also offer DIAL, our on-line technical support service to help you with the tough questions, and development courses that cover everything from using the dialog editor to memory management.

Find out how you can get your Microsoft Windows Software Development Kit. Pick up the phone and call (800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638. And we'll open the door to Windows.

The Microsoft Windows Software Development Kit includes:

- Dialog editor.
- Icon editor.
- Font editor.
- Resource compiler.
- Linker.
- MAKE (program maintenance utility).
- Symbolic debugger.
- Heap analysis utility.
- Sample programs.
- Windows libraries.
- Programmer's reference.
- Programmer's utility guide.

System requirements:

- 512K memory, DOS 2.0 or higher.
- Two double sided disk drives*
- Graphics adapter card.
- Hard disk recommended.

Microsoft Windows Software Development Kit

The High Performance Software.
Listing 4: The PostScript program to produce the conventional acuity test chart in figure 5.

\%
\% Produce Figure 5
50 750 moveto
/width 250 def
/size width 6 div def
/cuberoott2 2 1 3 div exp def
width 2 div size -1.5 mul rmoveto
/showrow
/psave
/FontDirectory /Sloan known
{/Sloan findfont size scalefont setfont}
{/Helvetica findfont size 1.8 mul scalefont setfont}
ifelse
dup stringwidth pop -2 div 0 rmoveto
show
restore
/size size cuberoott2 2 div def
@ size -2 mul rmoveto
def
(Z C H) showrow (Q S H) showrow
(H K N) showrow (O Z S) showrow
(D H C) showrow (C N O) showrow
(K D V) showrow (S H O) showrow
(D R H) showrow (K N Z) showrow
(R H K) showrow (C H O) showrow
(N K S) showrow (C Z R) showrow
(S O K) showrow (D N C) showrow
(C H D) showrow (R O H) showrow
(S N D) showrow (O C H) showrow
(Z H K) showrow (C V H) showrow
(V N Z) showrow (S C K) showrow
(N D C) showrow (C N D) showrow
(O V N) showrow (Z H V) showrow
(H S K) showrow (D N O) showrow
(N S K) showrow (D C Z) showrow
(V H K) showrow (S K V) showrow
(V R K) showrow (V R S) showrow
(K Z N) showrow
showpage

Some Linotype print shops will charge extra or refuse to run jobs that take many hours. For example, figure 4a took seven hours on an L300, and I was told, "Never again!" However, the Adobe books give no information about timing. Some of my jobs take only slightly longer to run on the Linotype than on the LaserWriter; others take much longer.

I find that I can get an upper bound on Linotype run time by measuring how long the job takes on the LaserWriter and multiplying by the factor by which the number of pixels will be increased. For instance, if the job takes 1 minute at 300 dpi on the LaserWriter, it might take up to (2540/300)² = 72 times as long (i.e., 72 minutes) at 2540 dpi on the L300. This is because the LaserWriter (and the LaserWriter Plus) and the Linotype L100 and L300 all use the Motorola 68000 microprocessor.

However, I've just learned that Linotype plans to upgrade the ROMs in all the Linotype L100 and L300 printers to PostScript version 42.5 soon. Apparently, this upgrade speeds up printing a lot, ranging from a factor of 1.5 times faster for printing noncached fonts to a factor of nearly 20 for bit-mapped images.

An odd deficiency of PostScript, given that it is a device-independent page-description language, is that it lacks any general device-independent way to describe the size of the page. To obtain nonstandard page sizes on the Linotype printers (up to 11.7 inches by 25.7 inches at 1270 dpi, less at 2540 dpi), you need to use a special operator, setpageparams, which is documented in a supplementary manual for the Linotype printers (see the text box on page 188).

PostScript Displays
Sun Microsystems recently announced a machine-independent standard for windows, called NeWS (Network/extensible Window System), which is based on building a PostScript interpreter into each window. What is to be displayed in a window is transmitted to the window as a PostScript program. NeWS is good for networking because you can easily load into the window a PostScript program that will interpret other window protocols, such as MIT's X windowing protocol, thus making NeWS highly compatible. NeWS will be a boon to PostScript programmers because it will let them try out their programs quickly on any display supporting NeWS, without waiting for a printer.

Debugging
The error messages from PostScript are quite specific and helpful, particularly
Get the Courier HST™ 9600-bps modem. Then watch the rest of the world play catch-up.

USRobotics new high speed technology delivers over 1,000 characters/second on more dial-up phone lines. For less than $1,000.

The new Courier HST (High Speed Technology) dial-up modem combines four great ideas that add up to a new standard for personal computer data communications.

It's full-duplex, with high speed (9600-bps) and low speed (300-bps) channels—automatically assigning the high speed channel direction. This asymmetrical solution avoids the problems of echo-cancelling technology or inefficient half-duplex schemes.

The most powerful data signalling technique—Trellis Encoded Modulation—lets Courier HST achieve maximum speed over a much wider range of phone line conditions than other 9600-bps modems.

A unique error- and flow-control method allows Courier HST to send up to 1,100 characters a second over long distance phone connections.

It's familiar as any 2400- and 1200-bps modem. Same features, commands and, in most cases, the same software. In fact, Courier HST automatically falls back to 2400, 1200 and 300-bps, connecting you with nearly all modems.

High speed. High accuracy. High value. And a two-year parts and service warranty. Get the USRobotics Courier HST, priced under $1,000. And watch the rest of the world play catch-up.

CALL 1-800-DIAL-USR
In Illinois (312) 982-5001

Yes, please send me your FREE 8-page book explaining the new Courier HST and today's new high speed technology.

Name
Title
Company
Address
City State Zip
Business Phone [__] 6/87 BY 2400F

USRobotics
The Intelligent Choice in Data Communications.
Listing 5: The PostScript program to produce the Pelli-Robson Letter Sensitivity test chart in figure 6.

```postscript
%! Produce Figure 6
50 750 moveto
/width 250 def
/size width 6 div def
width 2 div size -1.5 mul rmoveto
FontDirectory /Sloan known
{/Sloan findfont size scalefont setfont
/Helvetica findfont size 1.8 mul scalefont setfont
}ifelse
/dpi 72 0 matrix defaultmatrix dtransform dup mul exch dup mul add sqrt def
/screen dpi 16 div def
screen 127 gt [/screen 127 def] if
screen currentscreen 3 -1 roll pop setscreen /c 1 def
/setcontrast [1 sub neg setgray] def
/showrow /gsave
    dup stringwidth pop -2 div 0 rmoveto
    c setcontrast show
    grestore /c c 2 div def
    0 size -2 mul rmoveto
    def
    (N C R) showrow (C H V) showrow
    (Z R H) showrow (S H N) showrow
    (V D K) showrow (N K Z) showrow
    (S Z O) showrow (R D N) showrow
    (R Z S) showrow (Z R N) showrow
/showpage
```

Table 1: Print shops that output PostScript jobs on Linotype L300 printers.

<table>
<thead>
<tr>
<th>Company</th>
<th>Location</th>
<th>Telephone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ampersand Typegraphers</td>
<td>Toronto, Canada</td>
<td>(416) 422-1444</td>
</tr>
<tr>
<td>Cobb Typesetting</td>
<td>Cincinnati, OH</td>
<td>(513) 241-1146</td>
</tr>
<tr>
<td>Design North Inc.</td>
<td>Raone, WI</td>
<td>(612) 762-1320</td>
</tr>
<tr>
<td>Desktop Publishing Inc.</td>
<td>San Rafael, CA</td>
<td>(415) 258-0767</td>
</tr>
<tr>
<td>MacTypeNet</td>
<td>Livonia, MI</td>
<td>(313) 477-2733</td>
</tr>
<tr>
<td>National Colotype Corp.</td>
<td>New Berlin, WI</td>
<td>(414) 784-8960</td>
</tr>
<tr>
<td>ProTypography</td>
<td>Chicago, IL</td>
<td>(312) 266-8973</td>
</tr>
<tr>
<td>Typesetting Services Corp</td>
<td>Providence, RI</td>
<td>(401) 421-2264</td>
</tr>
</tbody>
</table>

because all the PostScript operators do extensive type and range checking of their arguments and give out a specific error message if they don't like them.

PostScript has several “print” commands that are helpful in debugging. They do not put ink on paper—instead, they send a message back to the host. (To put ink on paper, you say showpage.) One especially useful print command is ==, which will print whatever is on the top of the stack. What’s great about that is that anything can be put on the stack, even a compiled procedure, and == will uncompile it and print it out for you. It took me a while to realize just how useful printing is, since the Adobe books don’t say much about debugging. Incidentally, there are bugs in the PostScript interpreters (version 38.0) resident in the PostScript printers. For example, I discovered that stroking very large arcs can get a PostScript printer so confused that all subsequent jobs fail with a limit-check frame device error until you reset the printer. Adobe has acknowledged the bug and promises to fix it in a future release of the PostScript interpreter. The PostScript operator version returns a string with the version number of the interpreter.

Wrapping Up
You might find four periodicals interesting. Don Lancaster’s “Ask the Guru” column in Computer Shopper has clever ideas for nonobvious uses of PostScript and the LaserWriter. Colophon and Graphic Perspective, glossy newsletters illustrated with fancy PostScript illustrations, often include program listings. Finally, the PostScript Language Journal will publish its first issue about the time you are reading this.

PostScript isn’t really meant for programming, but it is very handy as a programming language to create images from mathematical descriptions. It offers various ways of describing your image, a high degree of machine independence, the possibility of debugging on moderately priced, widely available printers, and final printing at 2540 dpi for $10 a page. PostScript makes it easy to produce images on paper that would be difficult or impossible to produce otherwise.

REFERENCES