A Versatile TEX Device Driver

Minato Kawaguti
Fukui University, 9-1, Bunkyo-3, Fukui, 910 Japan
Internet: kawaguti@i1mps1.fuis.fukui-u.ac.jp

Abstract

A new TEX DVI driver was developed for Canon's LaserShot family (supported by LIPS III) and its twin, the LBP family (supported by CaPSL III/IV) laser-beam printers. High grade "publisher quality" TEX output using a conventional simple (that is, non-PostScript) laser-beam printer has been achieved: [1] four categories of fonts, (i) pk, (ii) printer-resident scalables, (iii) scalable fonts for emulating typesetters, and (iv) PostScript Type 1 format, can be used; [2] missing pk fonts may be generated automatically from either mf files or PostScript fonts; [3] various character ornamentations, such as: (i) gray scale inking, (ii) filling the glyphs with patterns, (iii) character outlining, (iv) drop-shadowing, (v) character shading, and (vi) black/white reversing, have been incorporated; and [4] insertion of figures into TEX documents through encapsulated PostScript files is possible despite the fact that non-PostScript printers are used.

Why non-PostScript Printer Driver?

A new TEX DVI driver was developed for Canon's LaserShot family (supported by LIPS III and IV control language) and its twin, the LBP family (supported by CaPSL III/IV control language) laser-beam printers. The main objective was to achieve high grade "publisher quality" TEX output with a conventional simple (that is, non-PostScript) laser-beam printer.

To be sure, the existing DVI drivers for PostScript printers, *e.g.*, Tom Rokicki's *dvips*, give excellent output. Nevertheless there exists, particularly among academic and research communities where people rely heavily on TEX, the persistent demand for a DVI driver with superior performance applicable to non-PostScript printers as well.

From the standpoint of TEX users, the majority of the pages of documents to be printed daily are in 'pure' TEX format. PostScript figures within a document, if any, occupy only a small fraction of the pages. In these circumstances, 'native' control language of the printers is equally or better suited, offering certain advantages in economy, and possibly in the printing speed and prolonged service life of the printer to the extent that complexity of the control software is simpler.

TEX is widely accepted as a prevalent norm for the generic means of exchanging documents across the computer networks. In the due course of matured sophistication of the TEX usage, however, the fine quality typesetting demanded by some documents is by now somewhat contradicted by the underlying implicit premise that the recipient is assured of reproducing the received documents exactly as the original copy the sender dispatched. This confidence, one of the characteristic virtues of TEX from the earliest days, can be reaffirmed much further by letting any available printer print practically any TEX document, irrespective of whether fancy typefaces are used, or some figures are embedded within the pages. All in all, it would be beneficial to the entire TEX community if we could enhance printing capabilities of all types of laser-beam printers.

Current Printer Environment

The current average TEX printer environment might be surmised through a recent episode of an international conference. The organizing committee of *Computing in High Energy Physics '92*, which was held at Annecy, France, in September 1992, announced to all the authors of the contributing papers that the camera-ready copy for the proceedings be submitted by typesetting with the network-distributed LATEX style file designed for the proceedings, which supposedly unified the font of the entire proceedings to Times-Roman and its derivatives.

Of the 197 papers actually published in the proceedings, a quick visual inspection reveals that those which conform to the Times-Roman or its near look-alikes were 130, while the remaining 67 were evidently alien to the specified font. The fact that only 66% could attain the style unification depicts

the still haunting problem of availability of fonts other than the Computer Modern.

With regard to non-alphabetic languages of which Japanese is one, the issue of fonts is more serious. The significant cost overhead associated with the Japanese PostScript fonts, and also the penalty of speed of PostScript printers in the past in dealing with these fonts, could have influenced the current proliferate use, among TEX users, of non-PostScript printers equipped with resident scalable fonts for the language.

Extended Functionality

The major objective of the development of the DVI driver to be described (henceforth denoted as the *Driver* for short), therefore, is to bestow the non-PostScript printer with the following extended features so that it can cope with the variety of unorthodox needs likely to be encountered in printing TEX documents written by some of the demanding users.

Figures. Insertion of figures in a TEX text should be performed in the standard manner. The figure, described in a separate file in the PostScript form, should be allocated at the specified location in the prescribed size. The image figure based on a bitmap file should also be included in the TEX document.

Fonts. It should be possible to mix the fonts in a multitude of formats, including pk fonts and standard PostScript fonts.

Ornamentation. For broadening the area of TEX application, it is highly desirable to add ornamental power of decorative expression beyond the standard TEX capabilities. Among others the following features are particularly noteworthy:

1. character highlighting for instruction manuals;
2. inclusion of half-tone shaded boxes for generating various forms;
3. artistic fonts for cover pages of books and manuals;
4. distinct taste or appeal of formal letters through the choice of adequate fonts; and
5. eye-catching special effects design for posters.

Basic Driver Action

The way the *Driver* prints the information in a DVI file can be summarized as follows:

1. The *Driver* translates the information in a DVI file into the 'native' control codes of the printer.
2. To insert a figure, the content of the specified PostScript file is interpreted by a slightly modified version of GNU's *Ghostscript* interpreter as a filter.
3. It deals with a large variety of fonts by obtaining the glyph data from the files on a multitude of media, including compact discs.
4. Various kinds of character ornamentations are performed for special effects while printing by TEX. The *Driver* receives the command specific to the printer through the 'special' command of TEX.
5. It manages the buffer memory of the printer while downloading the glyph data.
6. It generates the missing pk fonts, if necessary, from the mf files, or from PostScript Type 1 fonts.
7. It can maintain the size of the dynamic font directories below the upper bound specified by the system administrator.

Applicable Fonts

Four different kinds of fonts can be freely intermingled while typesetting:

1. pk fonts;
2. scalable fonts resident in the printer;
3. scalable fonts for emulating typesetters; and
4. scalable fonts in (encrypted) PostScript Type 1 format.

The *Driver* determines which font data among the actually implemented ones in the host computer or in the printer are to be used for each font specified in the DVI file, by looking at the font table which describes the corresponding relationship between them.

pk fonts. There are three categories of pk files, reflecting how they are generated:

1. pk fonts generated from mf file by METAFONT,
2. pk fonts created from PostScript font (typically Type 1 format font), and
3. pk fonts which have no corresponding generic data file mentioned above.

These fonts are located in three separate font subdirectories for ease of maintenance.

The standard TEX fonts, including Computer Modern, Computer Concrete, and AMS Euler, belong to the first category. The *Driver* presumes that the pk file of this category is always available. If found missing, the *Driver* immediately trips to the automatic font generation procedure to be described later.

The pk fonts of the second category are those generated mostly from the Type 1 format PostScript fonts. If the *Driver* discovers that the needed pk font of this category is missing while the corresponding Type 1 format file exists, the automatic font generation process is activated.

The pk font of the third category does not have its associated generic file. Some of the Japanese character fonts in wide use belong to this category.

The pk glyph files in the directories corresponding to the first two categories may be erased, if necessary, during the autonomous management of the disk space, whereas the files in the third category should never be erased.

Resident scalable fonts. The printer has some number of fonts as resident scalable fonts. When two optional font cards are mounted to the LIPS version of printer, for example, the *Driver* utilizes at least nine scalable fonts simultaneously, namely Dutch, Swiss, Avant Garde, Century Schoolbook, Bookman, Zapf, together with three standard Japanese typefaces, that is, Mincho, Square Gothic, and Round Gothic.

Scalable fonts for emulating typesetters. The font glyph data described by Bézier curves in conformance with a subset of PostScript language constructs are also available. They are derived from the proprietary fonts originally developed in Ikarus format for some of commercial typesetters. (As such they are not in PostScript font format.) Although this type of font is not necessarily in wide circulation, there are occasions when one wishes to simulate the typesetter with a local printer. Publishers and authors might be able to enjoy improved mutual communications.

Outline font data are sent to the printer after translating into the control language of the printer.

PostScript Type 1 format fonts. The fonts in this category can be used in two different ways:

1. The first method converts the data into pk format in advance, as described before, and then uses it as the pk font.
2. The second method converts the content of the Type 1 format data into a sequence of plain PostScript commands by means of the decryption algorithm as specified by Adobe Systems, Inc., and then sends it to the printer just as described for the plain PostScript fonts.

The advantage of the first method is its printing speed. It is best suited for routine use or for printing a document of significant volume. Since *xdvi* can display any pk font on the X-window screen, this scheme permits brisk and efficient previewing as well. In contrast, the second method sports the advantages of a scalable font. For example, gray scale inking is possible only when the second method is used.

Gray Scale Inking

Whereas in standard documents there is no particular need for printing glyphs with anything other than solid black ink, the introduction of gray scale inking to TEX offers the following merits:

1. It adds slightly more artistic flavor to typesetting, while balancing optical reflectivity of the entire page by making a glyph of a significantly large size font look milder.
2. Controlled level of emphasis may be given to each character of a string.

Glyphs of scalable fonts may be filled with any of the hundred grades of gray scale, ranging from pure white to solid black, with or without superimposing its contour line.

Since the inking on the glyph is treated logically as opaque, a glyph of lighter inking may be superimposed on top of a darker background.

Filling with Patterns

The printer under consideration is equipped with a primitive command which fills the local region enclosed by a closed circuit such as one element of a glyph. The filling is not limited to a homogeneous gray. Any of the 64 textural filling patterns the printer can generate, in addition to the 100 homogeneous gray scales, may be equally applicable. In what follows it should be understood, therefore, that the word 'filling' refers both to gray scales and to textural patterns without any distinction.

Other Character Ornamentations

For printing instruction manuals covering subjects associated with the screens of computer terminals, it is customary to reproduce the highlighted character strings displayed in reverse video mode faithfully on a printed page. The *Driver* can perform the character black/white reversing for this purpose.

Character shading may be a milder alternative of emphasizing character strings. Any of the filling patterns may be specified for the shading.

Drop shadowing (shadow characters) may offer the effect of somewhat artistic flavor. This can be accomplished by adding an arbitrary depth and pattern of shadow to each character.

To create various kinds of forms, the *Driver* may apply shading to a rectangular area.

Insertion of Figures

The way figures are inserted into TEX documents assimilates the manner adopted by other existing DVI drivers for PostScript printers.

Any file in encapsulated PostScript format (EPSF) is qualified as a figure file. To insert a figure, a new TEX macro is called in, which requests the designated file name, and as an option the figure size as well. The *Driver* modifies, if necessary, the width and the height of the figure independently so that the size matches exactly that specified by the macro parameters. The calculation of the magnification factors is based on the data of the original figure size contained in the header portion of the figure file.

As a contingency measure, the *Driver* accepts as well plain PostScript files which are not in EPSF. Since no rescaling is performed in this case, it is the user's responsibility to adjust beforehand the figure size to the parameters specified in the macro parameters. This extra option is to safeguard dedicated TEX adherents from the confusion in discriminating between the two PostScript formats.

A bitmap pattern, such as a dumped-out copy of an X-window screen, may also be included into a document. In most cases, however, bitmap data converted to EPSF is preferred because of the auto-scaling feature mentioned above.

Automatic Font Generation

If the *Driver* discovers that any of the pk fonts specified in the DVI file are missing, it generates the missing pk font automatically, provided that (1) the corresponding mf file exists, or (2) the font table registers the font to be generated from the corresponding PostScript Type 1 font and the PostScript font file actually exists.

In the first case, the *Driver* first activates *virmf*, which is a tool of METAFONT, to generate the pk file on the spot from the mf file, and then resumes the suspended printing work by using the newly created font, in the same way as *dvips*, and also *xdvi*, do in a similar situation.

In the latter case, it creates the pk file automatically with the help of GNU's *Font Utilities*, and then comes back to the suspended job again just as it does while working with mf files.

This feature, which helps reduce the chance of failure resulting from lack of fonts, will be found effective when printing jobs are queued across the network.

Processing Speed

The speed of printing is as important as the quality of output from the utilitarian point of view in daily service. Since the printer's internal buffer memory for storing font glyphs is limited to typically 6 MB in size, the *Driver* gives higher priority to the more frequently used fonts. The selection algorithm for the font registration checks the font size, the classification of the nature of font, and the statistics on the frequency of request in the past.

The speed of printing is greatest when the *Driver* uses the fonts which have been registered in the printer memory through the previous font downloading.

Because of the sheer volume of font data, printing Japanese documents is much more critical compared with the case with alphabetic fonts. Each Japanese font consists of 7285 characters. Without an adequate policy for the selection of font registration, precious resources will be used up quickly by fonts with less urgent needs, thus resulting in serious degradation of the performance.

Since the interface of the printer in use by the host computer is limited either to Centronics format or to RS232C, time spent in transmitting the glyph data governs the overall printing speed. Therefore it is extremely time-consuming to transmit each of the glyph data of fonts in frequent use from the host computer without storing them in the printer memory.

So as to boost the printing speed, the glyph data of the resident scalable fonts may also be registered into the buffer memory. While using multiple Japanese fonts, however, there occurs occasionally a situation when the amount of font data exceeds the volume of the buffer memory. In these circumstances, use of the resident fonts without registration can be a good compromise because it eliminates at least the time loss due to the glyph data transfer.

Compatibility

To eliminate system dependencies as much as possible from TEX source files while using the newly added extended features (gray scale inking, pattern filling, and character ornamentation), a style file which contains the related macros definitions is included in the (LA)TEX source files. Under the premise that these features are universally welcome

by a broad spectrum of the audience, it is hoped that TEX source files can attain a certain degree of system independence when the other printer drivers eventually incorporate their respective style files sharing the common format for these macros.

Evaluation

The *Driver* has been tested successfully with all models of laser-beam printer which were available for evaluation: Canon's B406S, A304E, A404E, and B406G. All of them are supported with the LIPS control language.

The measurement with the printer controlled by a 32-bit RISC chip at 600 dpi internal resolution (B406G) indicates that the printing speed reaches nearly the speed governed by the printer's paper handling rate for typical TEX documents without figures.

The average time needed to create a new pk font from a PostScript Type 1 font is about 20 seconds with a typical workstation, Sony's NWS-3860 (CPU: R3000 at 20MHz). Incidentally, it took slightly less than 19 hours to generate, in a single batch, seven standard sizes (10 pt and its assorted six magnifications up to \magstep5) of the pk fonts each for the entire 500 typefaces contained in *URW TypeWorks* compact disc. The generated 3500 pk files occupy 31 MB of disk space.

A full-page sample output is reproduced in the Appendix. To demonstrate the *Driver*'s ability of font handling and character ornamentation, a somewhat chaotic melange of fonts in various formats and sizes have been selected. It is hoped that the base font size as 20pt, or in part \magstep3 of 10pt, is large enough to make the finer detail of decorative features of the original 600 dpi output reproducible after printing.

It is emphasized that no PostScript processing is involved in generating this sample page. It simply generates the intrinsic command codes of the printer.

Implementation and Availability

The *Driver* is written entirely in C language. The total amount of source files of the current version is 190 Kb, or 8570 lines, excluding files associated with fonts data. It can be compiled using GNU's *gcc* compiler.

The standard TEX environment of Unix is the minimum prerequisite for porting the *Driver*. With this basic environment, printer-resident scalable fonts with ornamenting capability, together with pk fonts, can be used. In addition, GNU's *Ghostscript* has to be available in order to include figures in EPSF format. Likewise, GNU's *Font Utilities* is needed to deal with PostScript Type 1 fonts.

A revised version, which is currently being developed with improved portability and ease of maintenance in mind, will facilitate easier implementation on various Unix platforms, including 4.3BSD, OSF/1 and SunOS. This version, after subsequent field tests, will be available as free software by accessing, through anonymous ftp, the directory

```
/ftp_services/TeX_driver
```

at the address

```
i1nws1.fuis.fukui-u.ac.jp (133.7.35.53).
```

Conclusions

With the DVI driver for non-PostScript printers, the following assets have been realized:

A variety of fonts can be incorporated, including pk fonts, resident scalable fonts of the printer, and PostScript Type 1 fonts. As a part of performance evaluation, the *Driver* has been put into daily service incorporating, among others, 3500 pk fonts (500 distinct typefaces, each in 7 sizes) generated from a single CD-ROM of PostScript Type 1 format fonts.

Æsthetic features and visual appeal, essential in certain kinds of documents, can be realized conveniently with the knowledge of TEX alone. Character ornamentation, such as filling glyphs with a pattern, drop-shadowing, black/white reversing, and outlining, can be achieved without recourse to other technology, such as PostScript.

The insertion of figures by means of an encapsulated PostScript format file has been realized. The identical format to that used in PostScript printers has been acknowledged for compatibility.

Acknowledgements

The author acknowledges Ken-ichi Sugimoto, Atsushi Takagi, and Tetsuto Sakai for their effort in coding most of the software as part of their graduate study. He thanks Canon Corp. for offering printers for field evaluation. *URW TypeWorks* compact disc, used as the major source of PostScript Type 1 fonts, was supplied by Seaside Software Inc., Chigasaki, Japan. Special scalable fonts for emulating typesetters were supplied by Heidelberg PMT Co., Ltd., and also by Dainippon Printing Co., Ltd., both in Japan.

The *Driver* is indebted to various existing tools in one phase of its operation or another: *Ghostscript* and its adaptation to LIPS printers, *dvips*, *dvi2ps*,

Font Utilities, and METAFONT. Appreciations to their respective authors should be due in letting their tools be widely available.

Bibliography

Adobe Systems Inc. *Adobe Type 1 Font Format*, Addison-Wesley, 1990

Berry, Karl, and Kathryn A. Hargreaves. "Font utilities" version 0.6, 1992

Cooper, Eric, Bob Scheifler, Paal Kvamme, Håvard Eidnes, Mark Eichin, Paul Vojta, Jeffrey Lee, Donald Richardson, *et al.* "xdvi on-line manual", 1992.

Deutsch, L. Peter. "Ghostscript 2.6.1", 1993.

Knuth, Donald E. *The METAFONTbook*, Addison-Wesley, 1986.

Rokicki, Tomas. "DVIPS: A TEX Driver".

Sagiya, Yoshiteru, Hiroshi Ishii, Hajime Kobayashi, Ryouichi Kurasawa, and Hisato Hamano. *Japanese TEX Technical Book I*, ASCII Corp., 1990.

Sample Output

Selection of solid black, 50% gray scale, and outlined pure white inking at 30 pt: **Cheltenham Ultra** *Italic*. Outline may be superimposed to the gray glyph. The logo TEX uses one of 64 filling patterns for 50 pt **Bauhaus**. *[Font: Avant Garde]*

Character B/W Reversing is indispensable in expressing the inverted video display on the CRT screen.

Character Shading works for mild emphasis or printing forms.

Line Widths of Outline specified are 2pt, 4pt, and 6pt. *[Font: Century Schoolbook]*

Some shadowing examples: **Shadow + Patternfill**, **Shadow + Shading**, **Shadow + Outline**, 15% Right-Below, and 6% Down.

Character ornamentation shown above can be applied to the scalable fonts, irrespective of the font size. *[Font: Bookman]*

Exotic fonts like script styles go just as well with TEX: Freestyle, Avitus, Englische Schreibschrift, Commercial, Phyllis Initials, Park Avenue, Murray Hill, ***Brush Script,*** *to name a few. Japanese fonts, like* 内蔵 明朝体 *or* 輪廓を 描く, *in arbitrary size can be used and ornamented.*