editor. The format also allows easy extension to include local or additional functionality.

Conclusions

The TEX graphics macros presented above provide all the capability currently, and for the near future, required to include publication quality line art within TEX source files. They also include all the required capability to seriously begin experimenting with the direct inclusion of halftone images within TEX source files. Including publication quality halftone images within TEX source files requires both further development of digital 'halftoning' techniques and the general availability of higher resolution output devices.

The Challenge

The Challenge is for the macro gurus to implement the above TEX graphics macros and for the METAFONT gurus to generate the required fonts. I'll be happy to consult on the graphics aspects of the development, to test the results and to implement the filter program from .sdf format files to TEX.

Aerospace Engineering Department
United States Naval Academy
Annapolis, MD 21402, USA
E-mail: dfr@usna.navy.mil

References

1. Rogers, David F. and Adams, J. Allan, *Mathematical Elements for Computer Graphics*, 2nd Edition McGraw-Hill Book Co., New York, 1990, Appendix A. (Available in August 1989.)
2. Rogers, David F. and Rogers, Stephen D., A Raster Display Graphics Package for Education, IEEE Computer Graphics & Applications, Vol. 6, No. 4, April 1986, pp 51–58.
3. Rogers, David F., *Procedural Elements for Computer Graphics*, McGraw-Hill Book Co., New York, 1985.
4. Knuth, Donald E., *The TEXbook*, Addison-Wesley Publishing Co., Reading, MA, 1984.
5. Knuth, Donald E., Digital Halftones by Dot Diffusion, ACM TOG, Vol. 6, pp 245–273, 1987.
6. Jarvis, J. F., Judice, C. N., and Ninke, W. H., A survey of techniques for the display of continuous tone pictures on bilevel displays, Comput. Graph. Image Process., Vol. 5, pp 13–40, 1976.
7. Computer Graphics Metafile for the Storage and Transfer of Picture Description Information (CGM), ISO 8632–Parts 1 through 4: 1987; also, ANSI/X3.122-1986.
8. Arnold, D.B., and Bono, P.R., *CGM and CGI*, Springer-Verlag, Heidelberg, 1988.

A Portable Graphics Inclusion

Bart Childs
Alan Stolleis
Don Berryman

One of the serious limitations of current TEX usage is the lack of a portable inclusion of graphics. We propose a means of making this possible. It will require a little discipline on the creation of drivers; the procedures outlined herein should make it straightforward to add this to existing drivers.

Graphics inclusion has been a part of the drivers from the Computer Science Department at Texas A&M since soon after the initial release of the earliest QMS-QUIC drivers. In these cases it required significant positioning by the user and therefore became dependent upon the particular printer the document was destined for. This violated the intent of the dvi file, namely being device independent. Our previous graphics inclusion was much like the "Ph.D. with a screwdriver" concept.

It is my (Bart Childs) opinion that in spite of the beauty, power, and widespread use of PostScript, it is not a suitable answer for TEX output. There are two reasons for this opinion:

- A convenient manner for incorporating fonts from METAFONT is not yet in the public domain. *PostScript downloading of bitmaps is inconvenient at best!*
- The size of PostScript files is inordinately large, and use of the system in networks both clogs the network and makes the printing of documents happen at a fraction of the rated speed of the printer, especially if you use Computer Modern fonts.

These comments should not be taken to imply that the immense contribution of PostScript is not appreciated. I think that PostScript is in severe need of a binary mode. Another question of relevance for *de facto* standards is, will it be good for three-dimensional graphics? Many other questions need to be answered before we should treat it as a standard.

We need only a few elements to enable portable graphics inclusion. These elements are:

1. A standard template for the allocation of the size of the graphical area in the TEX document.
2. A standard means of putting the size of the graphical area in the dvi file.
3. A standard means of communicating the "name" of the file containing the graphics.

A typical line of TEX to cause the inclusion of a graphical element would be

```
\figinput{name}{horiz}{float}{caption}
```

The four arguments are:

`name` - The name is reasonably straightforward. Notice that it does not have an extension. File names could be the biggest problem in portability between dissimilar systems.

`horiz` - This positioning command would have one of three options: `\left`, `\center`, or `\right`. Text could flow around the graphics as well but that should be handled outside this consideration. We have borrowed this idea from Tom Reid's comments about positioning around figures.

`float` - This argument should be `\tied` or `\float`, which tells TEX to put the inclusion in a `\vbox` or an `\insert`.

`caption` - The caption should be optional. If present, it should include all required font changes.

There is a graphics standard which has been approved by ANSI and ISO. ANSI X3.124-85 and ANSI X3.124.1-85 is the Graphics Kernel System (GKS). The GKS standard defines how graphics interfaces function and how GKS data files should be stored. The computer graphics interface, `CGI`, is a program that provides user support for the creation of graphics. The Graphics Kernel System Metafile, `GKSM`, is simply an audit trail of the commands used by the `CGI` to create a graphic image. Within the GKS standard, graphic images have associated `cgm` files. The acronym comes from "computer graphics metafile" and is defined by ANSI X3.122-86 or ISO standard 8632. The `cgm` is a device independent file that decribes a graphic image. The `cgm` is similar in nature to the `dvi` files of TEX. The `GKSM` is different from the `cgm` in that the `cgm` is a definition and translation of the final graphic image. The `cgm` will not have all the steps used to produce the picture.

Most vendors deliver reasonable support for a GKS environment or it is available from third party vendors for common systems. Data General (DG) provides support to its end user community in a variety of ways, one being with the Comprehensive Electronic Office-Drawing Board (CEO-DB). Programming support is also provided for ANSI F77, PL/1, and C languages. DG also provides extensive GKS support to output devices. These include DG's graphics peripherals, PostScript devices, the HP family of printers, and others.

Our third author created a utility which inputs `name.cgm` and creates a file `name.siz`. Our second author has rewritten the original utility in CWEB and made extensive changes. This output file contains text like:

```
% This file created by cgmsize Version 0.1
\hsize %(width)
3.000000in %(height)
% Graphics file:  name
% Date of creation:  01/06/89
% Graphics designer:  Anonymous
\figdrafttrue
% vsize is estimated.
```

Lines 1, 4, 5, 6, and 8 are commentary. We intended that the `\figinput` macro would or would not place the word "DRAFT" in the margin based on the contents of the seventh line. If the `cgm` file does not exist, the utility will prompt for lines two through eight. The second line shows the default value if an answer is not given to the width prompt.

In addition to creating a properly sized and placed `\vbox` for the inclusion of the graphics, the `\figinput` macro should furnish a `\special` to communicate the name of the file containing the graphics. We use a line like:

```
\special{copy{name}}
```

Notice that the filename does not include an extension; remember the goal is device independence! The actual graphics file name should have the appropriate extension furnished by the driver depending upon the intended printer. We suggest the following three character (or shorter) extensions:

ps	PostScript.
qic	QMS QUIC engines.
lbp	Vanilla Canon engines or nearly compatibles.
jep	Hewlett Packard Laser Jet and compatibles.

One additional burden must be placed on the `\figinput` macro. Some printers will require placement of the "cursor" at the top left corner of the box for the graphics, some will require it to be at the bottom right; some will require the width of the graphics; etc. We propose that the `\vbox` have a `height` as indicated by the `siz` file. To enable device independence, the contents of the `\vbox` should contain:

1. An `\hrule` with the width of the box, height of 1sp (scaled point), and zero depth.
2. An `\hrule` with the height of the box, zero depth, and width of 1sp.
3. A `\vskip` back to the top left corner of the box.
4. The above `\special`.

Drivers should be modified to notice rules that have 1sp vertical and horizontal spans and to remember the nonzero dimensions in separate registers. This gives all drivers the necessary information to position themselves correctly and to copy the appropriate data file. We recommend passing the current position (the `\vskip` in item 3) at the top left because it agrees with the coordinate system of `dvi` files in some sense. We also understand that many would prefer adding two arguments to the `\special`'s copy, the height and depth. If this is done, we strongly recommend it be done in points only. This should be a decision by the standards committee.

The following is an outline of how we would create graphics and incorporate them in TEX documents:

1. Create a GKS file by use of utilities or applications that use GKS. We use CEO-DB to create a graphics image.
2. Have GKS output a `cgm` file. In our case with CEO-DB we would create a `cgm` file from the CEO environment.
3. Run `cgmsize` to create the appropriate file needed by TEX using `\figinput`.
4. Have a utility that creates the appropriate printer file.
5. We also have to run a utility that properly prepares the graphics output file for the desired printer. This utility will strip off characters at the beginning and end of the file, and reset the printer for TEX.
6. Run TEX.

After outlining this prototype system, we realized that the format would be appropriate for `dvi` inclusion as well. We might want to create a figure or table using LATEX, PICTEX, T2D4, or other macro package that uses lots of memory. A `dvisize` utility could produce an appropriate size file. A `dvimerge` file could then be run when needed to produce a "complete" `dvi` file.

We are forwarding this to Robert M^cGaffey and his group working on printer standards. We will be pleased to furnish this and the appropriate sources to all interested parties. Please respond with comments to the first author and copy Mr. M^cGaffey.

Output Devices

TEX Output Devices

Don Hosek

The device tables on the following pages list all the TEX device drivers currently known to TUG. Some of the drivers indicated in the tables are considered proprietary. Most are not on the standard distribution tapes; those drivers which are on the distribution tapes are indicated in the listing of sources below. To obtain information regarding an interface, if it is supposed to be included in a standard distribution, first try the appropriate site coordinator or distributor; otherwise request information directly from the sites listed.

The codes used in the charts are interpreted below, with a person's name given for a site when that information could be obtained and verified. If a contact's name appears in the current TUG membership list, only a phone number or network address is given. If the contact is not a current TUG member, the full address and its source are shown. When information on the drivers is available, it is included below.

Screen previewers for multi-user computers are listed in the section entitled "Screen Previewers". If a source has been listed previously under "Sources", then a reference is made to that section for names of contacts.

Corrections, updates, and new information for the list are welcome; send them to Don Hosek:
Bitnet `U33297@Uicvm`,
Internet `U33297@Uicvm.Uic.Edu`
(postal address, page 3).

Sources

ACC Advanced Computer Communications, Diane Cast, 720 Santa Barbara Street, Santa Barbara, CA 93101, 805-963-9431 (DECUS, May '85)

Adelaide Adelaide University, Australia

The programs listed under Adelaide have been submitted to the standard distributions for the appropriate computers. The PostScript driver permits inclusion of PostScript files in a TEX file. The driver is described in *TUGboat, Vol. 8, No. 1.*

AMS American Mathematical Society, Barbara Beeton, 401-272-9500 Internet: `BNB@Math.AMS.com`

Arbor ArborText, Inc., Bruce Baker, 313-996-3566, Arpanet: `Bwb@Arbortext.Com`

ArborText's software is proprietary and ranges in price from $150 to $3000. The drivers for PostScript