Graphics

Integration of TEX and Graphics at the Pittsburgh Supercomputing Center

Phil Andrews

Our Graphics Environment

The Pittsburgh Supercomputing Center is one of five NSF national Supercomputer Centers established to help scientific researchers. Our main machine is a CRAY YMP8/24 (8 processors, 24 million 64 bit words of memory) running UNICOS, the CRAY version of UNIX. We also have a large number of DEC machines running both VMS and ULTRIX, other general purpose computers such as a HARRIS HCX/UX and special purpose computers such as an ARDENT TITAN. Presently we have approximately 1000 users, the great majority of whom are remote and communicate with our center over networks such as the Internet.

The center was established in the summer of 1986 in the enviable position of being able to design our overall graphics system from scratch, having no compatibility requirements. We decided to standardise on the CGM (Computer Graphics Metafile) format for picture storage and acquired only graphics packages supporting that format, e.g., DISSPLA from CA-ISSCO, DI-3000 from Precision Visuals and the NCAR Graphics package from the National Center for Atmospheric Research. If possible we adapted other packages, such as MOVIE.BYU, to produce CGM format files.

While in 1986 CGM was an emerging standard for picture storage, it is now both an ANSI and ISO standard and is solidly ensconced as the preëminent format for the description of two-dimensional graphical data. There are several interpreters available for the display of CGM files on various output devices, but in general they can only display a small subset of the CGM elements and are specific to the CGM files produced by that vendor's graphical packages. In addition they may be proprietary, and, if purchased, we could not redistribute them to our remote users. In order to mitigate the problems of network access, we encourage our users to generate CGM files on our machines, and then ship the CGM output home to their host machines where it can be viewed in a more interactive manner.

For these reasons, and because we wanted to support any available output device, we decided to write and maintain our own CGM interpreter, called GPLOT for General PLOTting program. GPLOT now processes all of our users' CGM files, independently of their origin, and is used in-house for the production of video animations (approximately 10 hours of animations in the last 12 months) which we mail to users. We support numerous output formats, including PostScript and QMS (QUIC) laser printers, several types of workstations (via UIS, X11, or CGI interfaces), numerous Tektronix and other terminals, GKS devices and video output via a Peritek frame buffer. We are continuously updating the number of devices supported. GPLOT will accept either binary or clear text format CGM files and will convert between the two.

Integration with TEX

In designing the GPLOT system, I wanted to address one of the outstanding problems in computer related information interchange: the problem of text and graphics integration. I chose the TEX typesetting system for this purpose, partly because of its popularity and capabilities, and partly because of familiarity. In the late 1970's I ported first the old Pascal version and then the WEB version to both TOPS-10 and VMS and wrote DVI processors for Versatec and Tektronix output devices, and (later) for QMS laser printers.

I removed the DVI interpretation part of that program (GTEX), together with the font manipulation system, and integrated it into emerging CGM software to form our GPT system. Although GPLOT and GTEX are separate commands, they are part of a completely integrated system with over 90% of code in common. Any CGM file processable by GPLOT can be included in any TEX file by GTEX, and GPLOT uses the standard TEX font files (in PK format) for the textual parts of CGM files on output devices with limited textual capability.

Naturally the TEX interface to the CGM graphics routines is by way of a `\special` command, but many different formats are possible. With over 1000 users from differing backgrounds, documentation and training can be a significant problem, and, with this in mind, I decided to make the command format for the `\special` command identical to what the user would type at the command level. That is, if (under VMS) the user would type

```
gplot/dev=ps/pag=3/x_size=4.5/y_size=4.5 foo
```

to instruct GPLOT to process the CGM file `foo.cgm`, extract the third page only and produce output for a PostScript printer scaled to fit

a 4.5 by 4.5 page, then inside a TEX file the corresponding \special command would be

```
\special{gplot/pag=3/x_size=4.5/y_size=4.5 foo}
```

Note that the \special command itself is device independent, the output device being specified on the GTEX command line that caused the interpretation of this particular DVI file. Any device specification inside a \special command will be ignored. The current page position in the DVI file will become the origin for the included graphics page. Any number of pictures from any combination of files can be included in an individual TEX page. However the effect of overlays is completely device specific.

TEX processing specifics

GTEX and GPLOT use the same command line parser, with identical sets of options and supported devices; if the device can use downloaded fonts (e.g., PostScript or QUIC) then I use that capability (only the required characters are downloaded). Expanded versions of the character descriptions from PK format files are cached either internally in memory, or externally in an indexed file. For any TEX file of several pages or larger, the font processing is normally a small part of either the CPU or I/O requirements, and I do not attempt to use native fonts for TEX output. If the device cannot download fonts then I convert character references to cell array calls (the CGM raster operation). Rules naturally map to CGM rectangle calls. One interesting result of this mapping is that GTEX can convert DVI files into CGM files; these CGM files, however, will be partly resolution-specific with each character represented by its bitmap. A proposed addendum to the CGM standard adds segments (recallable descriptions), and as soon as this becomes official I will use this operation rather than cell arrays for character representation. There are many devices that do not support downloadable fonts but do have segment support, and of course the associated CGM files will be much smaller.

Problems

GPLOT and GTEX are wholly in the C programming language and run under both UNIX and VMS, but with different command interfaces. That means that presently the \special command syntaxes are also different under each system, an unacceptable format. As our users become more accustomed to UNIX I hope to move away from the VMS format for \special commands, supporting the UNIX format on both systems. However I plan to continue to support the VMS format under VMS for our users who are unconcerned with portability.

Many CGM files use colour tables for their output, then use indices into this table to designate colours. When two such pictures, with distinct colour tables, are included on a single TEX page there may be clashes, depending on the capabilities of the output device. I intend to eventually do an automatic conversion from indexed to direct colour in these cases. The difficulty is in deciding when this is necessary.

TEX-specific graphics

In some cases what is really required is the capability to process simple graphics at runtime, rather than the inclusion of complex preprocessed graphics. In this case the difficulty lies in organisation rather than implementation. If there is no requirement for interaction with the location of TEX elements then a simple clear text CGM file can be written and included. For more complex requirements I have allowed (for experimentation purposes) a simple \special command that allows access (at runtime) to the capabilities of any CGM element processor. This can be used to change text colour, produce lines linking TEX elements, fill polygons, etc. What is needed here is some set of simple \specials, preferably community wide. However it should not be thought that some such set can satisfy all graphics requirements. Many graphics packages have more manpower investment than the entire TEX system and cannot be easily simulated.

Availability

We are presently distributing GPLOT/GTEX freely via Internet (we don't write tapes), although it is copyrighted and we request that you do not redistribute any modified, spindled, folded or mutilated copies. Nor can you sell it, include it any package for sale, etc. For further information send mail to ANDREWS%CPWSCB@CLIPR.PSC.EDU.

◇ Dr. Phil Andrews
Pittsburgh Supercomputer Center
Mellon Institute
4400 Fifth Avenue
Pittsburgh, PA 15213
andrews%cpwscb@clipr.psc.edu