PBMtoPK — Converting bitmaps to fonts

Angus Duggan (`ajcd@uk.ac.ed.lfcs`)

21 September 1990

PBMtoPK.

PBMtoPK is a program for converting and collecting bitmaps into a PK format font, for use with TEX. PKtoPBM performs the inverse operation, extracting bitmaps from a PK format font.

PBMPLUS.

The PBM (Portable BitMap) format was created by Jef Poskanzer as a "lowest common denominator" format, which images could be translated to and from with ease. The bitmap representation is given in plain ASCII text, so that images can be mailed between different machines and networks without the usual problems of encoding and decoding. (The ASCII representation of the bitmap also allows you to create or alter images with a text editor, if no better tools are available.)

The PBM format is very simple:

```
P1
16 16
0000011111100000
0001100000011000
0010000000000100
0100000000000010
0100000000000010
1000110000110001
1000110000110001
1000000000000001
1000000000000001
1000000000000001
1000100000010001
0100010000100010
0100001111000010
0010000000000100
0001100000011000
0000011111100000
```

The image above is a smiley face (☺). The first two characters of the file contain the "magic number" of the file — in the case of a PBM file, this is "`P1`". The width and height of the image follow, seperated by whitespace (spaces, tabs or newlines), and then the image data, encoded as ones and zeros. The PBM format specification allows white space to appear almost anywhere in the file — the image need not have newlines between image rows, and the bits can be seperated by whitespace if desired. Comments can be included in PBM files, but most of the PBM filters will delete them from their output.

The PBMPLUS toolkit contains programs to perform many common image manipulation operations on PBM format images, such as cropping, cutting, pasting, rotating, shearing, converting between image formats, etc.. These programs are usually arranged as *filters*; they take the input image on the default input stream, and write their output to the default output stream. This organisation allows multiple image manipulations to be strung together in a pipeline, creating no unnecessary temporary files.

There are two other image formats which are supported by the PBMPLUS toolkit. These are the PGM (Portable GrayMap) format, and the PPM (Portable PixMap) format, which are lowest common denominator formats for grayscale and colour images respectively. These formats have the magic numbers P2 and P3, and a similar organisation of the image data. In addition, there are binary variants on the PBM, PGM and PPM formats, which are unsuitable for mailing. These variants have the magic numbers P4, P5, and P6.

Writing programs which use the PBMPLUS formats is made a lot easier by the PBMPLUS libraries, which provide functions for reading and writing PBMPLUS images in their entirety, or row by row.

The PBMPLUS toolkit has been ported to VMS and other systems.

Using PBMtoPK.

PBMtoPK and PKtoPBM are based on Tom Rokicki's PXtoPK and PKtoPX programs. They are written in C, and are linked with the PBM library. PBMtoPK can be used to incorporate bi-level images into TEX documents, by creating a font which contains the images used in the document as characters. Figure 1 shows an example graphic which was created in this way.

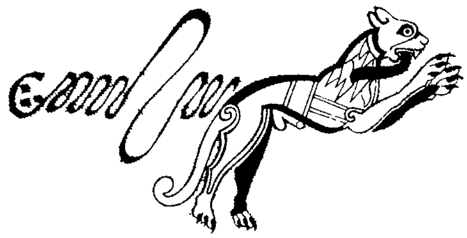

Figure 1: A scanned line-drawing converted by PBMtoPK

Some DVI drivers may have problems handling the large characters which are typically created by this process. This should not be a problem if the driver adheres to the recently-published draft Level 0 DVI driver specifications.

To create a simple font containing only one image (in the character position of "A"), a command similar to

```
pbmtopk example.300pk example.tfm 300 -s10 -c65 image.pbm
```

would be used. This command will create the PK font file `example.300pk` and the TFM file `example.tfm`, setting the resolution of the font to 300 dots per inch, and the design size of the font to 10 points. To include this image in the document, a font declaration should be put in the document prologue:

```
\font\imagefont=example
```

The font may then be selected, and the image can be typeset by placing the character "A" on the page. For example:

```
\begin{figure}[hbt]\centering
\mbox{\imagefont A}
\caption{A scanned image converted by PBMtoPK}
\end{figure}
```

The TFM and PK files will probably have to be moved to an appropriate directory (the exact directory names are site and DVI driver specific); DVI drivers and previewers may also require that the location of the PK and TFM files are included in option files or environment variables.

The major disadvantage of this method of including raster graphics in TeX output is that the user must know many details about the output device.

PBMtoPK provides a large number of options which can be used to alter information in the PK and TFM font files — the designsize, resolution, font parameters, and the coding scheme and font family name comments can all be reset. Options are provided to set the character's width, height, depth, and italic correction in the TFM file for each character (these need not be the same as the character's actual width or height in pixels), and the character origin and horizontal and vertical escapements in the PK file. Sensible guesses are made for any piece of information not supplied, based on the information which was supplied and the image size.

If multiple images are being included in one font file, or a font is being converted from another format into a PK font, the command-line options rapidly become unwieldy and may overflow the maximum command line size. A facility is therefore provided to read the character-specific options from a file, which can be prepared with a text editor, or automatically generated.

Ligature and kerning tables are not catered for at all by PBMtoPK — if these features are required in the font, the TFM file should be converted to a PL (Property List) file, edited by hand, and converted back into a TFM file.

The interface to PKtoPBM is much simpler; the name of the PK file from which glyphs are to be extracted is given on the command line, with a list of names for the PBM files mixed with switches which set the number of the next character to be extracted.

Getting PBMtoPK.

PBMtoPK and PKtoPBM are on the Aston archive, in the directory `[tex-archive.wherever]`. PBMtoPK was posted to Usenet group `alt.sources` in early August, and so should be found in any archive storing `alt.sources`.

Angus.