Halftone Output From TEX*

Dr. Adrian F. Clark
Department of Electronic Systems Engineering
University of Essex

August 11, 1995

1 Introduction

This note describes how image data can be converted to a form suitable for TEX and then typeset in documents. Pictures are prepared for TEX in a somewhat different way to normal text—the hard work is done outside the TEX input file; incorporating the image into the TEX document is then quite straightforward.

The image data must be converted from the binary representation used in Image Processing and Pattern Recognition work to a series of characters which TEX can map into pseudo grey-levels via a special font. A user-callable FORTRAN `SUBROUTINE` called `TEXPIC` is available for this purpose. `TEXPIC` converts a picture held in memory to a text file containing TEX input. (You can look at this file if you like, but it is unlikely to make much sense.) There are a few additional routines which can be used with `TEXPIC` to produce particular effects: for example, the picture may be plotted as a positive or negative image.

The special font for setting pictures is defined to have some 65 grey levels. This does not necessarily imply that all the grey levels are distinguishable, nor even that they are strictly monotonic on laser printers. However, the quality of the output is certainly adequate for run-of-the-mill technical reports.

An example of an image typeset with `TEXPIC` and TEX is shown in Figure 1. This is the "girl" picture from the image database distributed by the Signal and Image Processing Institute of the University of Southern California, which is widely used to demonstrate image processing techniques. This representation is some 64×64 pixels.

*Taken from the *VAX/VMS TEX User's Guide*

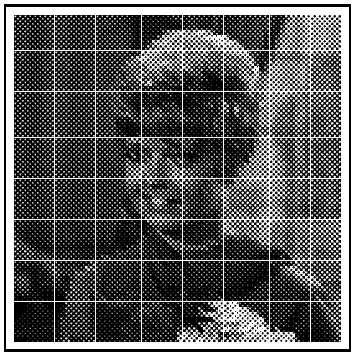

Figure 1: Example of an Image Typeset using TEXPIC and TEX

2 The TEXPIC Routine

To generate a file containing the image data in a form suitable for input to TEX, you simply invoke TEXPIC from a program. TEXPIC is called as follows:

```
CALL TEXPIC( ARRAY, M, N, FILE )
```

where ARRAY is a M × N REAL array containing the picture to be plotted and FILE is a quoted string or CHARACTER variable containing the name of the file to receive the data. If you omit a filetype ("extension") from FILE, TEXPIC will use .TEX. You can call TEXPIC as many times as you want from a program; a separate output file will be generated for each picture.

Due to problems with the controlling micro-processor in the standard LN03 printer, pictures with N $>\approx$ 64 will come out with white bands across them, rendering them effectively useless. However, this problem has been overcome on the LN03+ device, which can print images up to a full A4 page in size.

3 Associated Routines

TEXPIC normally maximises the contrast of pictures by determining the range of grey levels in the image and scaling the output to make use of them. This behaviour can be overridden by invoking the routine ZRANGE before TEXPIC. ZRANGE requires two REAL arguments, the minimum and maximum grey level values to be output:

```
CALL ZRANGE( ZMIN, ZMAX )
CALL TEXPIC( ARRAY, M, N, FILE )
```

where ZMIN $\leq$ ZMAX. TEXPIC then uses the supplied extrema, rather than values determined from the image, for scaling the data. Values which lie

outside these extrema are clipped. Hence, if `ZRANGE` is used with values which are outside the values of the pixels in `ARRAY`, the contrast of the data generated by `TEXPIC` is *reduced.* The converse is also true. The extrema set up by `ZRANGE` remain in effect until another call is made to `ZRANGE`, superseding the previous limits, or `ZAUTO` is invoked, which restores the default behaviour.

`TEXPIC` normally plots positive pictures—i.e., pixels which contain low values come out darker than pixels with high values. It can also produce negatives, however; you do this by invoking the routine `DONEG` before `TEXPIC`. Normal, positive picture-plotting is restored by calling `DOPOS`. As with `ZRANGE` and `ZAUTO`, `DOPOS` and `DONEG` remain in effect until explicitly cancelled. You can, of course, use the routines in conjunction to produce any required effect.

There is a limit, $M_{\max}$, to the number of pixels that `TEXPIC` can fit across a page of output. For pictures with `M` $\leq M_{\max}$, the data are written out exactly as supplied. However, when pictures have `M` $> M_{\max}$, the image is interpolated so that the data written out form a $M_{\max}$-pixel image. The same interpolation factor is used in both dimensions so that the aspect ratio of the picture remains correct. The interpolation technique used is a context-sensitive three-point bilinear method due to Smith.[1] This produces marginally better results in the vicinity of edges in the picture.

The default value of $M_{\max}$ is 256. You can alter this (for example, when producing pictures for a document with the text set in narrow columns) by invoking `TEXMAX` before `TEXPIC`, as follows:

```
CALL TEXMAX( MV )
```

This sets $M_{\max}$ to `MV`. Obviously, `MV` ≥ 2.

`TEXPIC` and the support routines are all to be found in the algorithm library, `ESE$LIB:ALG`. There is also a stand-alone program which you can use for images stored in the standard formats on disc. This program is `ESE$PROGRAMS:V159`. site specific

4 Inserting the Picture into Your TEX Document

This is quite easy, although there are a few steps in the process. The procedure you follow depends on which macro package you are using with TEX. Most people use "plain TEX," the default macro package, but LATEX is also in widespread use. If you use a customised TEX but don't know exactly what is different about it, try the plain TEX procedure.

[1] There is also a version which uses standard four-point bilinear interpolation.

Whichever version of TEX you use, there are a few things you should know. TEX must hold the whole image in memory, so the maximum image size is obviously limited to TEX's memory capacity. Since TEX was designed for typesetting rather than image hard-copy, its memory capacity isn't large enough to hold dozens of pictures, although it should be adequate for reasonable use.

`TEXPIC` tells TEX to use a special font, a *halftone* font, for setting the picture. You have to tell TEX to load this font. To do this, you type the following command at the top of your document:

```
\font\halftone=halftone
```

This works with both plain TEX and LATEX.

We shall now consider the procedure for telling TEX to typeset your picture. In the examples that follow, it is assumed that the picture produced by `TEXPIC` is in the file `PICTURE.TEX`.

4.1 Procedure for Use with Plain TEX

The first thing you must do is to create an input file for TEX which contains the document to be typeset. At the point where you want the picture to appear, you type:

```
\centerline{\input picture}
```

For TEXperts, the image is set as a single `\hbox`.

The picture can be put into a "floating" insert, which will cause the picture to be held in TEX's memory until there is enough free space on the page to hold it. For example, to set the picture in a `\midinsert`, the relevant TEX input would be:

```
\midinsert
   \centerline{\input picture}
\endinsert
```

This command sequence should be typed between paragraphs, when TEX is in 'vertical mode.' Typing `\goodbreak` immediately before the `\midinsert` may help TEX's page-breaking mechanism. (In practise, you'll probably want to add a title to the picture, too.)

There is one other thing you might want to do and that is to draw a border around the picture. You would do this by defining a TEX macro, `border`, as follows:

```
\def\border#1{\vbox{\hrule\hbox{\vrule\kern3pt
   \vbox{\kern3pt#1\kern3pt}\kern3pt\vrule}\hrule}}
```

You would then set the picture with

```
\centerline{\border{\input picture}}
```

instead of simply `\centerline{\input picture}`.

4.2 Procedure for Use with LaTeX

The most sensible way to plot pictures with LaTeX is in the `figure` environment (*not* the `picture` environment). This generates a "floating" figure, which usually surfaces at the top of the following page of output. The `\centering` declaration causes the picture to be centred in the page.

```
\begin{figure}
   \centering
   \mbox{\input picture\relax}
   \caption{Figure Title}
\end{figure}
```

The `\relax` following the filename in the `\mbox` command is required for LaTeX to know where the filename ends. To draw a border around your picture, you simply replace the `\mbox` command with `\fbox`.

5 Producing the Output

Having inserted the appropriate commands into your document, you run TeX in the normal way. If it exits with a "memory capacity exceeded" error message, you've either tried to set images which are too big or you're holding too many in memory at once. The only solution to this problem is to re-format your document. LaTeX users may be able to cure the problem by strategically inserting `\clearpage` commands into the manuscript.

You run `TeXview` or `TeXlase` on the TeX output file as normal, although the `TeXview` display bears little relationship to the grey levels produced by `TeXlase`. Note that you must use `TeXlase` version 10 to send output to the `LN03`, since earlier versions cannot read the halftone font file. (When you invoke `TeXlase` it outputs a short introductory message which includes its version number.)

The current version of TeX 2.0 has a larger-than-usual memory capacity, large enough for four 256×256 images to be held in memory. site specific