Figure 11: *Computer Modern Candy Stripe.*

Although this dependence on `grep` is outside the TEX-Metafont loop, its advantage is that you can pass more than numeric data to TEX. In the method I outlined previously, it was our job to keep straight which label text belongs to which label position. With `diagramf`, you combine the text with the position data in the Metafont file, so no further record keeping is necessary.

Using Diagramf

1. MF: create figure.
2. grep: strip log file.
3. TEX: run file to measure label text.
4. Return to beginning to resolve forward references.

Figure 12: *The steps in using* `diagramf`.

There is a further advantage to this method. After TEX typesets the label text, it can pass metric information back to Metafont. As a result, Metafont can revise the diagram to conform to the text. Figure 13 shows one result of this in practice.

There are three other packages users should be aware of. Both allow the placement of text at positions inside previously prepared PostScript figures. These diagrams can be produced by other applications like Adobe Illustrator, MatLab, Fontographer, and so on. Craig Barratt of Stanford has produced the `PSFrag` style for use with LATEX, while Jean Orloff of CERN the `PSBox` style, for use with either LATEX or plain TEX.

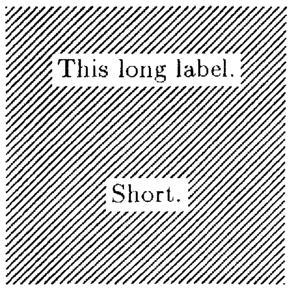

Figure 13: *Cutouts fit around their labels.*

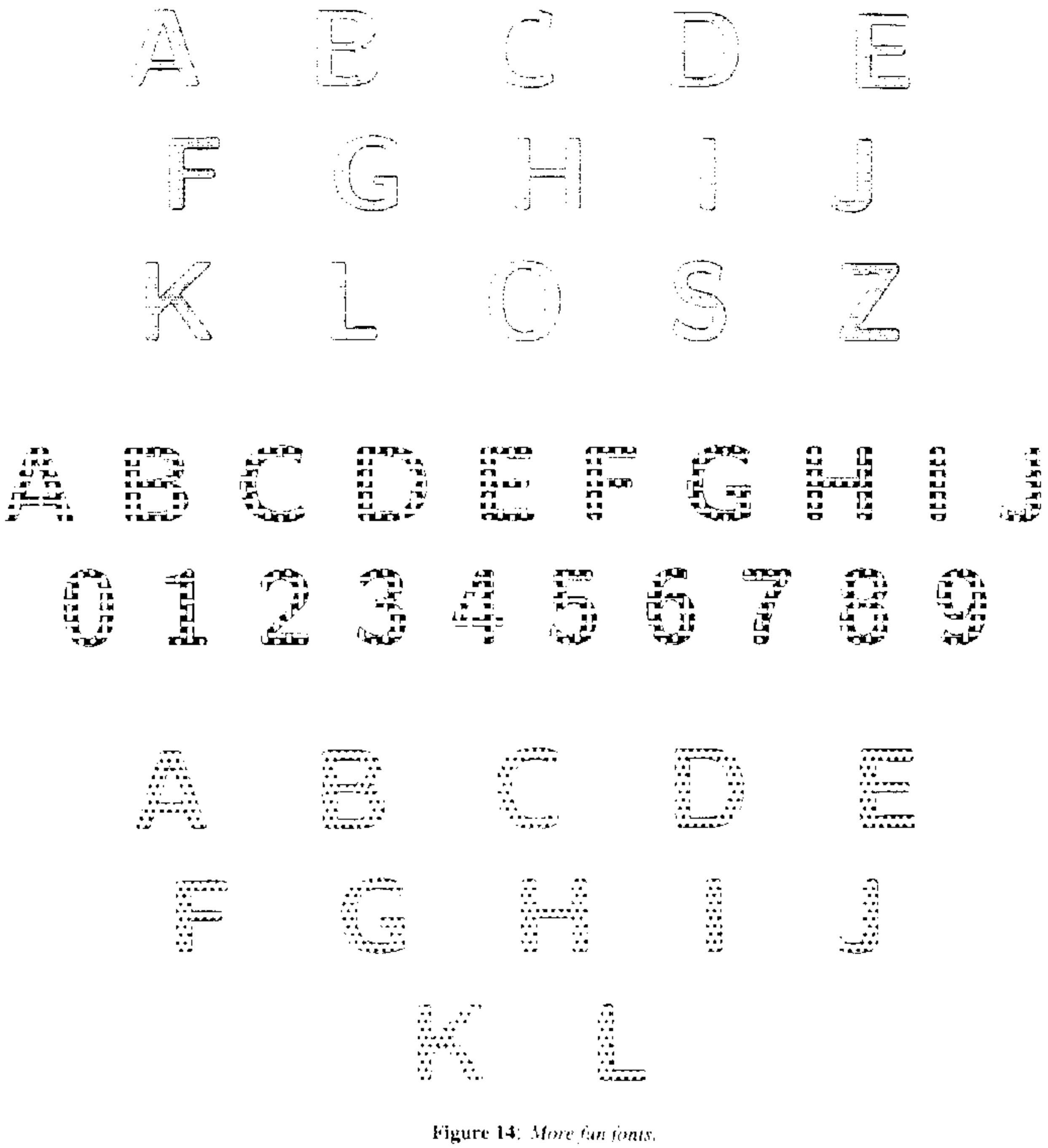

Figure 14: *More fun fonts.*

Just recently, I became aware of another such package that I'd like to mention. This is the `mfpic` macro package by Thomas Leatherum. It's not a terribly extensive package—the user manual TEXs out to four pages of `\magstephalf` text—but its product cycle is the inverse to that of `labtex` and `diagramf`. That is, you start with TEX code, and `mfpic` generates the appropriate Metafont files. I am excited about `mfpic` for a simple reason. Since all the `mfpic` commands are *TEX* commands, this package is therefore a perfect introduction to Metafont to those users who are too timid to tackle Metafont itself. Actually, since the pictures are created with `mfpic`'s TEX commands, you fit the labels into the picture using usual `\kern` commands rather than the special sleight of hand that I or Alan Jeffrey used. (Or so I gather from a very brief perusal of this package.)