Fetamont: An extended logo typeface

Linus Romer

Abstract
The logo font, known from logos like METAFONT or METAPOST, has been very limited in its collection of glyphs. The new typeface Fetamont extends the logo typeface in two ways:

- Fetamont consists of 256 glyphs, such that the T1 (a.k.a. EC, a.k.a. Cork) encoding table is complete now.
- Fetamont has additional faces like “light ultra-condensed” or “script”.

The fetamont package provides \LaTeX{} support for the Fetamont typeface. Both the package and the typeface are distributed on CTAN under the terms of the \LaTeX{} Project Public License (LPPL).

The following article presents some facets of the Fetamont typeface, explains important techniques and shows the history of the logo typeface.

1 Comparison with existing logos
The following picture shows the METAPOST and the METAFONT logos written in Fetamont (gray) and Taco Hoekwater’s Type 1 version of the logo font (outlined).

There are hardly any differences; only the “S” is significantly different, because its shape was changed by D. E. Knuth in 1997 (see section 6). The other faces of Hoekwater’s Logo are also very similar to their corresponding Fetamont faces. Widths and kernings may rarely differ by one unit (except for the “A” in Logo 9, which has a strange width).

A comparison with the METATYPE1 logo from Jackowski, Nowacki, and Strzelczyk (2001) shows virtually no differences as well.\footnote{I have never seen the original sources of the “Y” and the “1” but I think that my imitated “Y” and “1” are extremely close to the original.}

2 Comparison with the mflogo package
The control sequences defined in the fetamont package are analogous to those from the mflogo package (Vieth, 1999). \texttt{\LaTeX{}} \texttt{\LaTeX{}} \texttt{\LaTeX{}} produce the well-known logos of METAFONT, METAPOST and METATYPE1.

3 The many faces of fetamont
It is clear that thanks to the power of METAFONT the number of possible faces is endless. However, Fetamont comes in a mere 36 predefined faces. The suffixes of every face are schematically listed in the following table:

<table>
<thead>
<tr>
<th>Upright</th>
<th>Oblique</th>
</tr>
</thead>
<tbody>
<tr>
<td>r8 b8 h8</td>
<td>o8 bo8 ho8</td>
</tr>
<tr>
<td>r9 h9 h9</td>
<td>o9 bo9 ho9</td>
</tr>
</tbody>
</table>

Condensed Upright Condensed Oblique
| lc10 c10 | lco10 co10 |
| bc40 | bco40 |

Ultracond. Upright Ultracond. Oblique
| lq10 | lqo10 |

Script Upright Script Oblique
| lw10 w10 bw10 hw10 | lwo10 wo10 bwo10 hwo10 |

Anyone wishing to design a new face for Fetamont can do so by just redefining the parameters of \texttt{ffm10.mf} and saving the file under a new name.

3.1 Script faces
The Fetamont script faces make use of randomized paths that are drawn by a rotated ellipse pen to make it look more handwritten. They may be used for comics or children’s texts:

Since version 1.3, the OpenType versions of the script faces even support the “Randomize” feature.

3.2 Condensed faces
The titles in Knuth’s books show a variant of the logo font that blends with Computer Modern Sans Serif Demi-bold Condensed 40. So I decided to add this variant as Fetamont Bold Condensed 40 and let also a light and medium variant benefit from the condensation because it looked so good.

3.3 Ultracondensed Face
I went even a step further and created an ultracondensed face for Fetamont. The credits written on
movie posters are often typeset in such ultracondensed faces:

The most important thing in the programming language is the name. A language will not succeed without a good name. I have recently invented a very good name and now I am looking for a suitable language.

(This is said to be a quotation from D. E. Knuth.)

4 Spacing problems with the “S”

The original spacing of the letter “S” fits perfectly for the combination “OST” as in \textsc{MetaPost}. However, in combination with normal letters like “N”, the “S” is positioned too much to the right (see the left part of the following picture).

\textsc{NSN NSN}

Thus, I shifted the “S” a bit to the left (see the right part of the upper picture) and added corresponding kerning instructions for “OS” and “ST” to keep the spacing of the original logo intact.

5 Special techniques

5.1 Anchor pairing with \textsc{MetaFont}

In order to draw accented and other combined characters, it is helpful to use anchors. The concept of anchors is common in type design outside of the \textsc{MetaFont} world. However, anchors rarely have been seen in \textsc{MetaFont} up to now.

The idea is easy: Put an anchor at a given point in a base glyph and in the accent glyph; then overlay the two glyphs such that the anchors coincide, producing the pre-composed accented character.

stores anchors as ordinary points in arrays. Furthermore, the character picture is stored in another array at the end of each character. For combined characters, \textsc{MetaFont} looks at the different arrays and then overlays the base and the accent character using the macro \texttt{addto currentpicture}.

5.2 Kerning classes with \textsc{MetaFont}

Like anchor pairing, the concept of kerning classes is widely known but not frequently used in \textsc{MetaFont}. The reason for this is that \textsc{MetaFont} cannot natively write kernings for multiple characters at once. Hence, multiple kerning information has to be cached in arrays.

Let me illustrate the general idea of this caching with a fictional example:

5.2.1 Define kerning classes

It is clear that “OV” needs the same kerning as “DV”. But beware, “VO” needs a different kerning than “VD”! So generally there are two kinds of kerning classes:

- The first kerning class groups glyphs together that share the same shape to the right, like “D” and “O”
- The second kerning class groups glyphs together that share the same shape to the left, like “C” and “O”

The first kerning class is stored in an array as follows:

\begin{verbatim}
row 0 | 2 |
row 1 | 3 | 68 (D) | 79 (O) | 214 (¨O) |
row 2 | 2 | 86 (V) | 87 (W) |
\end{verbatim}

The 0th column is reserved for the number of columns (0th row) and the number of items in the corresponding row, because there is no straightforward way to determine the length of arrays or subarrays in \textsc{MetaFont}. Each row forms a first kerning class. The characters are stored as codes (for the sake of clarity, they are shown here as letters also).

The storage of the second kerning classes works analogously, in another array:

\begin{verbatim}
row 0 | 2 |
row 1 | 4 | 67 (C) | 79 (O) | 80 (Q) | 214 (¨O) |
row 2 | 2 | 86 (V) | 87 (W) |
\end{verbatim}

5.2.2 Kern the kerning classes

The classes are then kerned with the following commands:

\begin{verbatim}
addclasskern(“D”, “C”, 2u#)
addclasskern(“D”, “V”, -u#)
addclasskern(“V”, “C”, -u#)
\end{verbatim}

Linus Romer
This kerning information is stored in a large three-dimensional array, which has as many columns as characters:

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>6</td>
<td>67 u#</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>u#</td>
</tr>
<tr>
<td>79</td>
<td>6</td>
<td>67 u#</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>u#</td>
</tr>
<tr>
<td>86</td>
<td>4</td>
<td>67 u#</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>4</td>
<td>67 u#</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>6</td>
<td>67 u#</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>2u#</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>u#</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>u#</td>
</tr>
</tbody>
</table>

5.2.3 Write the kerning information

At the very end, the macro `writeligtable` writes all kerning information from the large three-dimensional array, row-wise, in a METAFONT-friendly way.

5.3 Producing outlines

The METAFONT sources have been converted to outline font formats like Type 1 or OpenType with a Python script. The script calls METAPOST to produce PostScript files for each glyph. These glyphs are imported by the `fontforge` module. Hosny (2011) previously used this technique to produce the outlines of Punk Nova. Because the glyph widths are lost in the import, the `tfm` module from the `mftrace` project is also needed (Nienhuys, 2006).

6 History of the logo typeface

- **1979/07/12**: Knuth (1979a) shows the first versions of the “Computer Modern” typeface and the METAFONT logo. The graphic below is the result of a conversion from the original “manfnt” sources (Knuth, 1979b), which were written in the obsolete METAFONT78.

**METAFONT**

Knuth used quite a thick circular pen. This “manfnt” also contains 8 pt, 9 pt and title versions of the logo typeface. However, the title font is just a magnified version of the 10 pt font.

- **1984/05/27**: The sources of the logo font are rewritten for the new METAFONT84 (Knuth, 1984a). The pen has become elliptic and thinner.

- **1984/09/09**: Second try for the new METAFONT84 (Knuth, 1984b). The characters “E” and “F” have rounded vertices.

**METAFONT**

- **1985/09/03**: Knuth (1985c) defines some “crazy shapes”. He then (Knuth, 1986) uses them to demonstrate randomized typefaces.

**METAFONT**

- **1985/09/22**: The logo typeface gets a slanted variant (Knuth, 1985b), a backslanted skinny bold variant (Knuth, 1985f) and an ultrawide light variant (Knuth, 1985d).

**SansMETA**

- **1989/04/22**: Knuth (1989) shows a new special weight to go with Pandora (see Billawala, 1989).
1992/01/16: Knuth (1992a) adds a demibold variant. This variant is needed for the title “Excerpts from the Programs for \TeX{} and \METAFONT” (Knuth, 1992b), which combines \textit{Computer Modern Bold Extended} with the logo font.

TEX and METAFONT

1993/03/23: Knuth (1993a) adds the characters “P” and “S” to the logo font such that one can typeset “\textsc{METAPost}” with it. Hobby (2009) states that both of the following logos are okay:

\begin{center}
\textsc{METAPost} MetaPost
\end{center}

1997/04/20: Knuth (1997b) changes the shape of the “S”. “It now sort of assumes that a ‘T’ will follow” (Beeton, 1998).

1997/09/30: The American Mathematical Society provides Type 1 versions of the logo font in the following styles: \texttt{logo8}, \texttt{logo9}, \texttt{logo10}, \texttt{logobf10}, \texttt{logosl10} (AMS, 2009). The “P” and the “S” are not included and the widths of the supplied glyphs are often wrongly rounded.

\begin{center}
\includegraphics[width=0.2\textwidth]{logo10}
\end{center}

1999/6/03: Taco Hoekwater releases new Type 1 versions of the logo font (Hoekwater, 1999). The conversion has been done in MetaFog, which is only available with \textsc{TrueTEX}. The “S” still has its old shape. The round endings are sometimes suboptimal (e.g. the “F” from \textit{Logo 10}) but all in all, the conversion is very clean.

\begin{center}
\includegraphics[width=0.2\textwidth]{logo10}
\end{center}

2001/03/04: Taco Hoekwater presents \textit{Elogo}, an extended version of the logo font, which covers the 7-bit \textsc{TeX} text encoding (Hoekwater, 2001). The font has been used in some of Hoekwater’s presentation slides (e.g. the following image is extracted from Hoekwater, 2007). The “S” still has the old shape.

\begin{center}
\includegraphics[width=0.2\textwidth]{logo10}
\end{center}

2001/09/26: At Euro\TeX{} 2001 the METATYPE1 logo is shown in the presentation of Jackowski, Nowacki, and Strzelczyk (2001). The newly designed characters “Y” and “1” are different to the “Y” and “1” in \textit{Elogo}.

\textbf{METATYPE1}

Funnily, the Euro\TeX{} 2001 conference logo also embeds an extended logo font (Pepping, 2001).

\begin{center}
\includegraphics[width=0.2\textwidth]{eurotex2001}
\end{center}

2011/06/02: For the documentation of another typeface, I wanted to write “mf2pt1” in the logo font, because I was heavily relying on this program back then. Having realised that there were other tools related to \METAFONT that could not be written in the logo font, I started extending the logo font.

\textbf{MF2PT1 MFLua MetaFog}

References


Linus Romer
Knuth, Donald E. “Letters for the METAFONT logo; first try with the new MF”. www.saildart.org/LOGO.MF[FNT,DEK], 1984a.
Knuth, Donald E. “Letters for the METAFONT logo; second try with the new MF”. www.saildart.org/LOGO.MF[FNT,DEK], 1984b.
Knuth, Donald E. www.saildart.org/METAFONT.MF[M,SY], 1985a.
Knuth, Donald E. “Fat version of METAFONT logo”. www.saildart.org/FLOGO.MF[M,SY], 1985d.
Knuth, Donald E. “Special font for the \TeX and METAFONT manuals”. www.saildart.org/LOGO.MF[M,SY], 1985g.


○ Linus Romer
Oberseestrasse 7
Schmerikon, 8716
Switzerland
linus.romer (at) gmx dot ch

Fetamont: An extended logo typeface