The making of a (TEX) font

TACO HOEKWATER   HANS HAGEN
Bittext, Dordrecht   Pragma ADE, Hasselt
info (at) bittext dot nl

Abstract

We want to introduce a new display font to the TEX community. The font is a digitization of a series of Duane Bibby drawings, commissioned by Pragma ADE. The digital version for use with ConTEXt is prepared by Bittext based on scans provided by Pragma ADE.

Figure 1: The first drawing

Figure 2: Rough design

Figure 3: Shapes were drawn on a grid, and refined over time

Introduction

At TUG 2003 in Hawaii, Hans Hagen met with Duane Bibby. Hans was looking for some small images to enliven the ConTEXt manuals. A cutout of a very early sketch can be seen in Figure 1, but it was soon agreed that consecutive drawings were going to be an alphabet.

Nothing much happened after that initial meeting, until the beginning of this year when Hans picked up the thread and got Duane started drawing. The alphabet quickly progressed. Starting in a rather naturalistic style like Duane's 'normal' TEX drawings, but later progressing toward a much more cartoon-like style, as can be seen from the drawings in Figure 2.

For ease of use, it was clear that these drawings should ideally become a computer font. Taco Hoek-
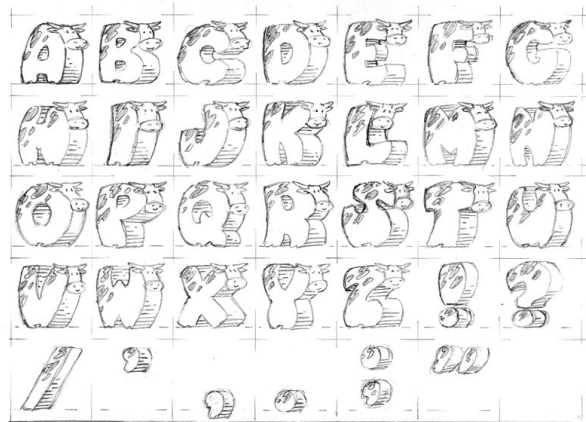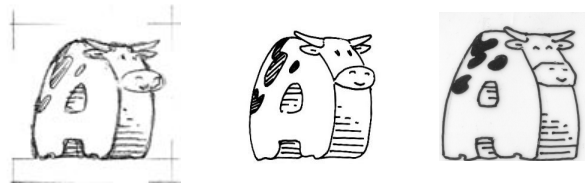
water agreed to take care of the digitization, and luckily the drawings were already prepared for that. As can be seen from the leftmost closeup in Figure 3, the cows are drawn inside a grid. This ensures that they are all the same size, which is a vital requirement for a font.

The center drawing in Figure 3 is a still rather roughly inked version of one of the in-between drawings (there were many). In this particular one you can see that the mouth of the cow was originally more or less oval, but in the final form (on the right) it became much more hexagonal.
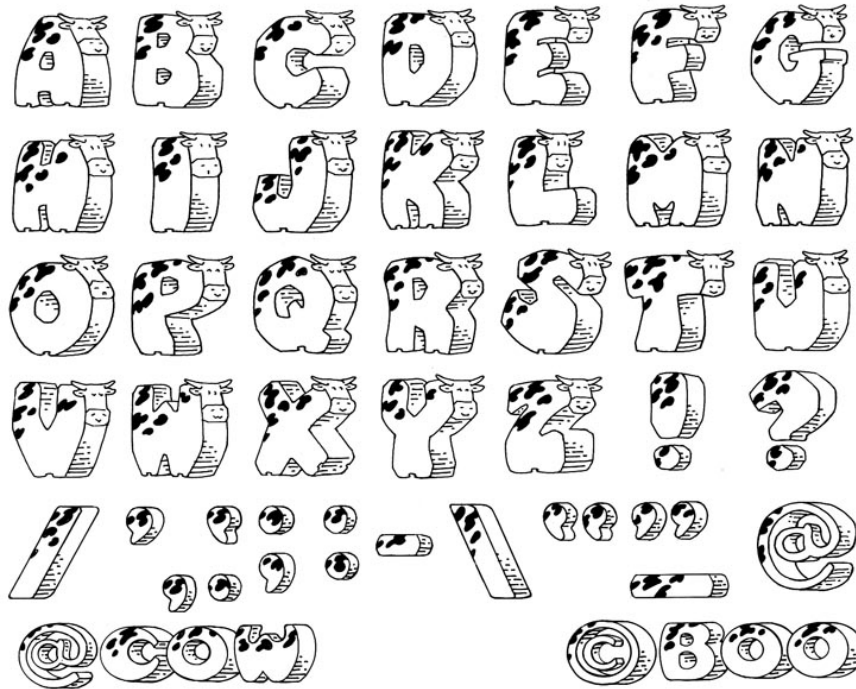
Figure 4: One of the original sheets, showing the alphabet and latin punctuation

Digitization

The original sheets were sent to Pragma ADE by regular mail in the beginning of March. Hans scanned the original sheets at 1200 dpi and then forwarded the images to Taco. There were four sheets in all, and one of them is shown in Figure 4. The other three contain a number of TEX-related logos and a few (mathematical) symbols.

Preparing the images

The first task in the preparation of the font was to create a set of bitmap images for use by FontForge's import command.

For this, the four sheets had to be cut up into many smaller pieces, each containing a single glyph for the font. This being intended as a decorative font, the character set does not even contain the complete ASCII range. Nevertheless, almost a hundred separate images were created.[1]

FontForge automatically scales imported images so that they are precisely one em unit high. After cutting the sheets up to pieces, the images therefore
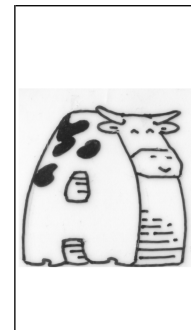
[1]It would have been nice if this step could have been done solely with free software, but the Gimp turned out to be incapable of handling the 75 megabyte PNG images for each of the four scanned sheets. Adobe Photoshop was used instead.
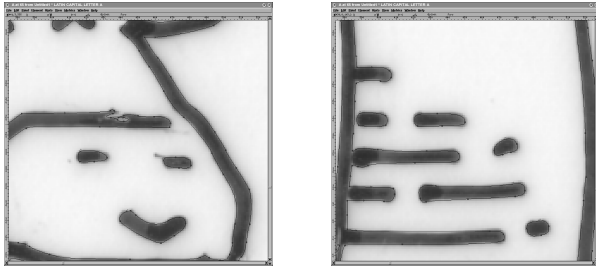

Figure 5: An imported bitmap images, with height adjusted

had to be adjusted so that they all had the same height. Without that, it would have been nearly impossible to get all the drawn lines in the glyphs the same width. Figure 5 shows the adjusted version.

Automatic tracing

The autotracer in FontForge, which is actually the stand-alone autotrace program, does quite a good job of tracing the outlines. But, interestingly enough, only at a fairly low resolution. At higher resolutions it gets confused and inserts more than a quadratic amount of extra points for each resolution increase. Based on empirical tests, the images were scaled to 40% of their

Figure 6: Close-ups of autotracer output

original scanned size, resulting in bitmaps that were precisely 1000 pixels high.

As was to be expected, the autotracer brought out many of the impurities in the original inked version, as you can see in the left image of Figure 6. Luckily, the number of places where manual corrections like this were needed was not so big to force us to reconsider the digitization process.

A more severe problem can be seen in the right-hand image of Figure 6. The drawings contain hardly any straight lines. For a font of this complexity, it turned out to be absolutely necessary to simplify the curves. Without simplification, the rendering speed in pdf browsers became unbearably slow. All of the near-horizontal stripes in the bellies were manually removed and replaced by absolute straights.

Hinting

The final stage in the font editor is to add the Post-Script hinting. A screenshot of a manually hinted letter is shown in Figure 7.

This part of the work is in fact turned out to be one of the largest jobs, because it is necessary to find a balance between two possible extrema.

On the one hand, if there are no hints at all, that results in nice small fonts that render quickly, but poorly.

On the other hand, if there is a lot of hinting information, that creates a much better appearance but it slows down the rendering. And sometimes extra hinting produces worse rendering than less hinting, especially with non-commercial renderers.

A middle ground can be reached, but unfortunately only by doing all hinting manually, and that took quite a lot of time.

Finishing the font

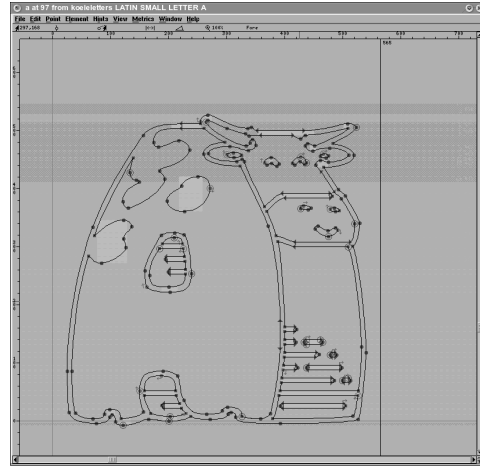The font was saved as two separate PostScript Type
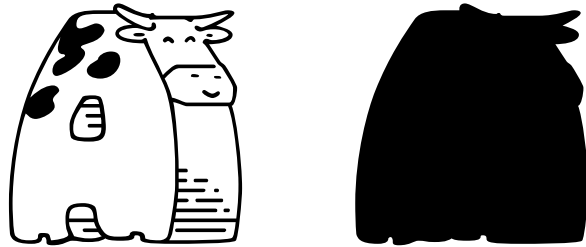

Figure 7: Finished outline, with hints

Figure 8: The final 'A'

1 fonts, one with the text glyphs and one containing the logo glyphs. The text font is named 'koeieletters', the logo font 'koeielogos'. 'Koeieletters' literally translates from Dutch to English as 'cowcharacters', but the word 'koeieletter' is also is used to indicate a really big character. Like in a billboard, for instance.

Eventually it turned out that we needed a second set of two fonts. Sometimes you want to have text in the cowfont but on top of a colored background. The background would then shine right through the hide of the cow and that was of course unacceptable. Hence, we also have the fonts 'koeieletters-contour' and 'koeielogos-contour'.
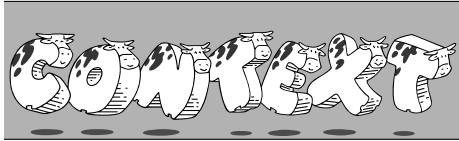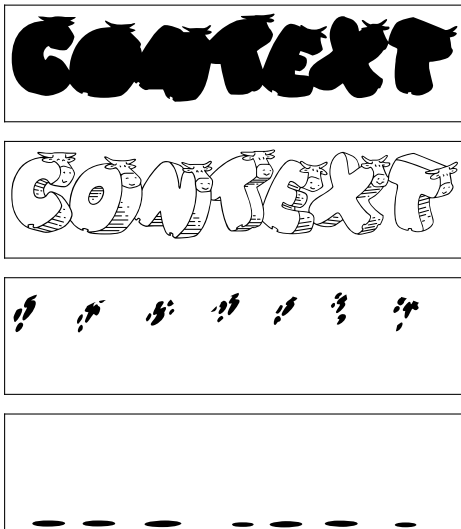
Figure 8 shows the final 'A', in the normal and the contour font.

Playing around

The original goal of this font was to enliven the Pragma ADE manuals. It would be a waste if we did not try to get the most out of the drawings provided by Duane, so we coded some rather silly effects in the font and its metrics. The final paragraphs of this article highlight a few of those.

Figure 9: The Pragma ADE logo

Figure 10: The ConTEXt logo

Figure 11: Four characters of the ConTEXt logo

Pragma ADE

The only lowercase symbols in the font are in the Pragma ADE logo itself, see Figure 9.

The ConTEXt logo

If you look closely at the ConTEXt logo in Figure 10, you can see that the shadows and the spots of the cows are drawn in different shades of grey (or colors). This is only possible because there are actually four characters involved instead of just the logo and the background contour, as shown in Figure 11.

Logo ligatures

The line on Figure 12 can be input directly as "Cows in ConTeXt", thanks to a handcrafted virtual font. This font contains a complete set of ligatures that

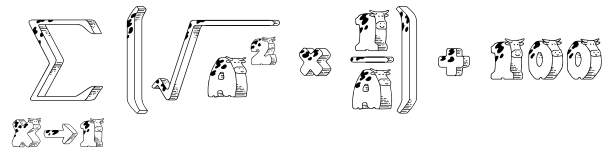Figure 12: "Cows in ConTEXt" typeset

Figure 13: Word logos

Figure 14: Cows in math mode

Figure 15: Old-style sheep

travel from 'C at the beginning of a word', past 'e following the temporary ligature C_o_n_T', all the way to 't at the end of word'. Such word logos are defined for TEX, ConTEXt, and MP, see Figure 13.

Mathematics

As is true for the collection of normal text glyphs, the math set is also not very extensive. But there are just enough math symbols to allow some example math formulas to be created. Virtual fonts make sure that the input is what you expect from TEX. See Figure 14.

Old-style Sheep

Seeing nothing but cows does tend to get boring after a while. To prevent the font from getting too predictable, we decided we needed some extra freshness. That is why the old-style numerals are actually sheep (see output in Figure 15):

```
I count \oldstylenumerals{10} sheep
```

Final words

The koeieletter fonts can be downloaded from the Pragma ADE site. The fonts, needed typescript file and macros are part of the standard ConTEXt distribution.