

MicroPress MathPrecise Font Technology

MicroPress

July 10, 2002

This version of the document has been reduced for online posting. Some examples and explanation has been omitted.

1 Overview

Beginning with $\text{V}\text{T}\text{E}\text{X}$ version 8.0, MicroPress starts shipping the new generation of Type1 fonts for TEX use. Unlike previously available Type1 fonts, the new MathPrecise by MicroPress are mathematically valid and are free from design and implementation errors which plagued all the previous implementations.

In this paper we will compare four implementations of CM/EC fonts:

- The “standard” AMS implementation of CM fonts, © 1991.
- The original MicroPress implementation of the EC fonts, 1996-1998.
- The Szabo TEX trace implementation, 2001.
- The Volovich SuperFont implementation, 2001.
- The MathPrecise implementation, 2002.

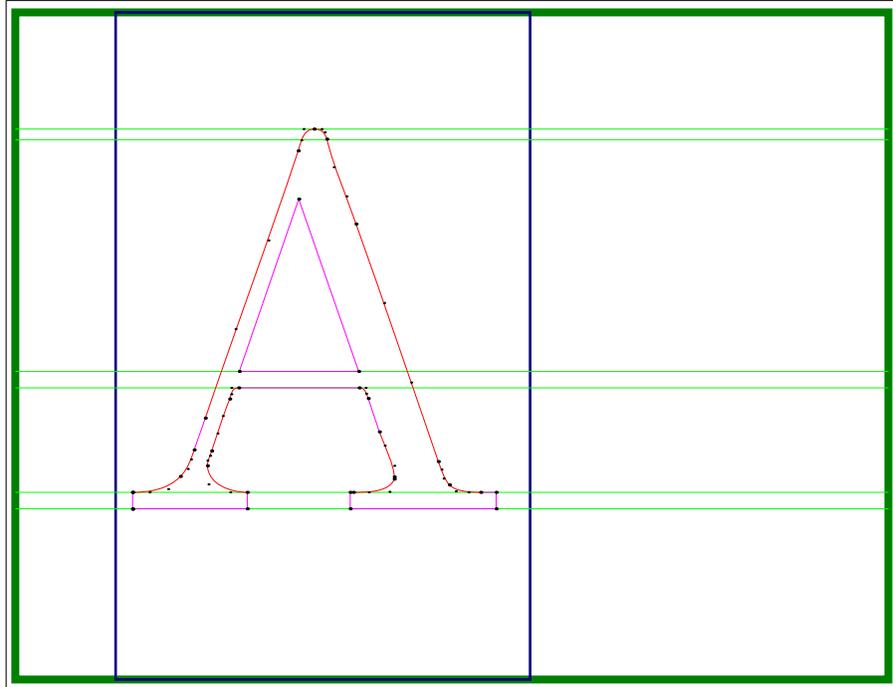
The first four implementations were done by an autotracer with additional manual fixing: none or almost none in the case of the TEX trace and “SuperFont’s”, and very extensive in the case of the AMS set or the old MicroPress’ EC set. No autotracing was involved in the MathPrecise version.

Bot TEX trace and SuperFont implementation were produced by the same autotracer, but the results are slightly different because of apparently different choice of the tracing parameters and postprocessing.

These two implementations are similar in providing very bad quality of the result; and we will start by looking at them.

1.1 The Letter A

The figure below shows the implementation of the letter A in the $\backslash\text{tenrm}$ font (the most commonly used font in the Computer Modern set):



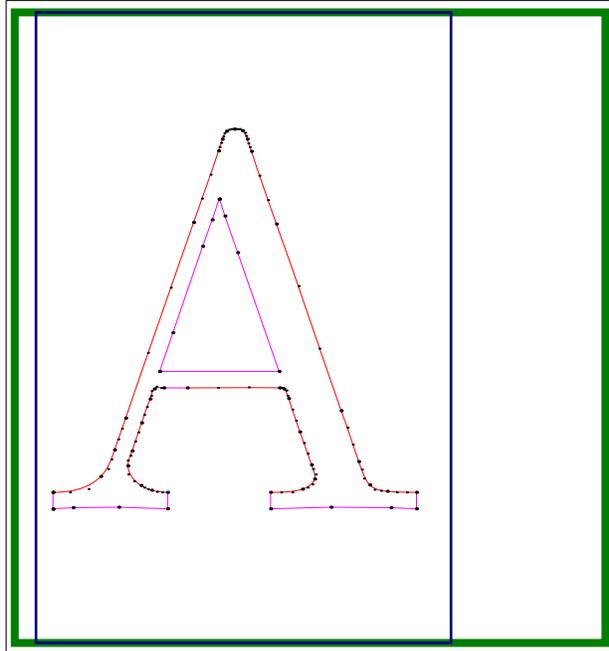
It is easy to see the deficiencies in the letter: the lines do not connect smoothly, the top of the letter shows a strangely shaped bump, and the flexes which should be present in the “legs” of the letter are simplified out of existence. Further, the sides of the letter, which are supposed to be straight lines, are actually curves here; with the left side not even approximating a line. Strangely, the inside edges the sides are lines; unfortunately, it is the outside ones that matter more. Finally, the connection of the horizontal bar to the sides is smooth on the bottom but sharp on top; this happens when a badly tuned autotracer cannot distinguish between sharp angles (usually, those that exceed > 90 deg) and arcs.

Many of errors present here can be automatically detected by a font editor, as shown here:

This image has been removed from the online copy of the document as to keep the document smaller

(red arrows show detected errors). Similar errors are present in nearly all glyphs in the “SuperFont set” and we will show a few below; but first lets examine how the letter “A” works in other fonts.

Here is the same glyph in the public domain T_EXtrace version:



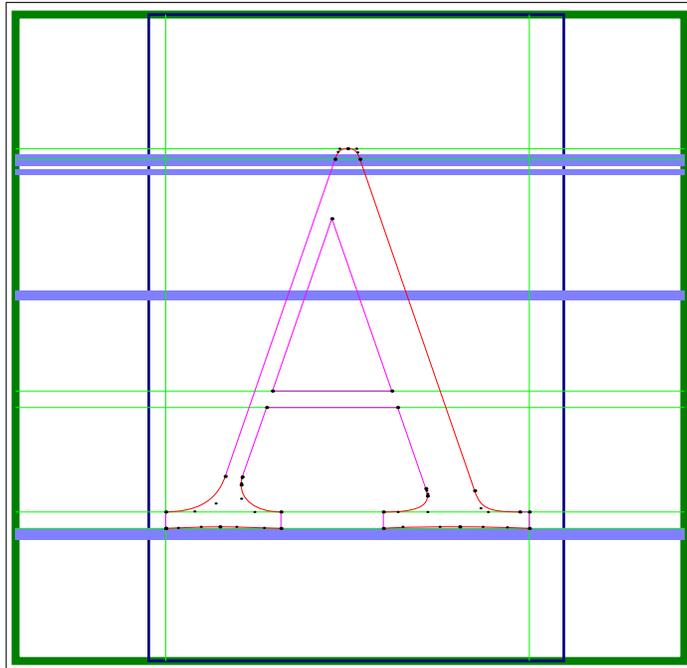
One major difference is the full absence of hinting of any kind; this, however, does not mean that the $\text{T}_\text{E}\text{X}$ trace version is inferior to SuperFonts since incorrect hinting is often worse than no hinting whatsoever.

It is apparent that the autotracer here has been set for more precise matching; this leads to larger number of points and correspondingly larger font size. The deficiency at top of the letter is “handled” by using six (6!) cubics to draw the top of the letter; in contrast, `cmr10` uses 3 and the MathPrecise uses 4.

Similarly, the rounding below the horizontal bar is present, but is less prominent here.

The flex is visible, but is not coded correctly.

In the standard AMS `cmr10` font we see



This is a much cleaner picture and the errors are not immediately visible. A Font Editor would show a couple present:

This image has been removed from the online copy of the document as to keep the document smaller

but these are less serious roundoffs (the malignant bump on the top of the letter is, however, visible) and are less likely to affect the appearance when the font is used in text.

One serious design error here is the absence of a proper BlueZone on top of the letter; this leads to sticking out top of the letter “A” when used together with other uppercase letters. This deficient hinting leads to

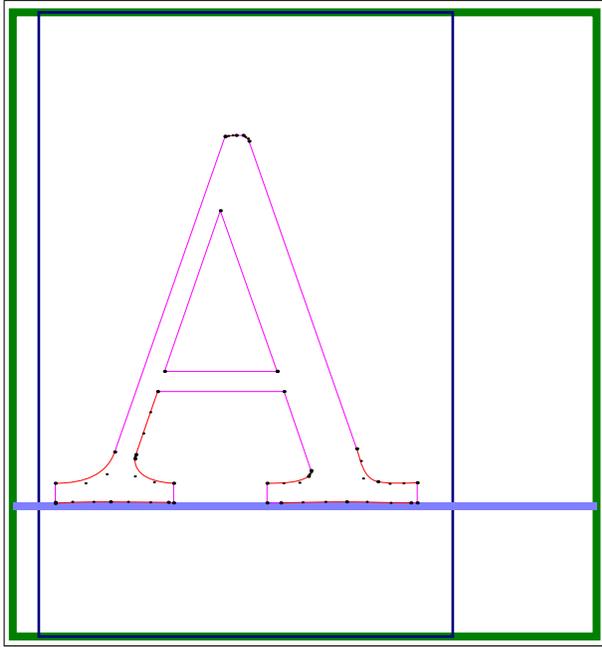
MAMA MAMA MAMA MAMA MAMA

where at some – but not all – sizes “A” appears higher than “M”.

Additional problems with the CM fonts lie elsewhere:

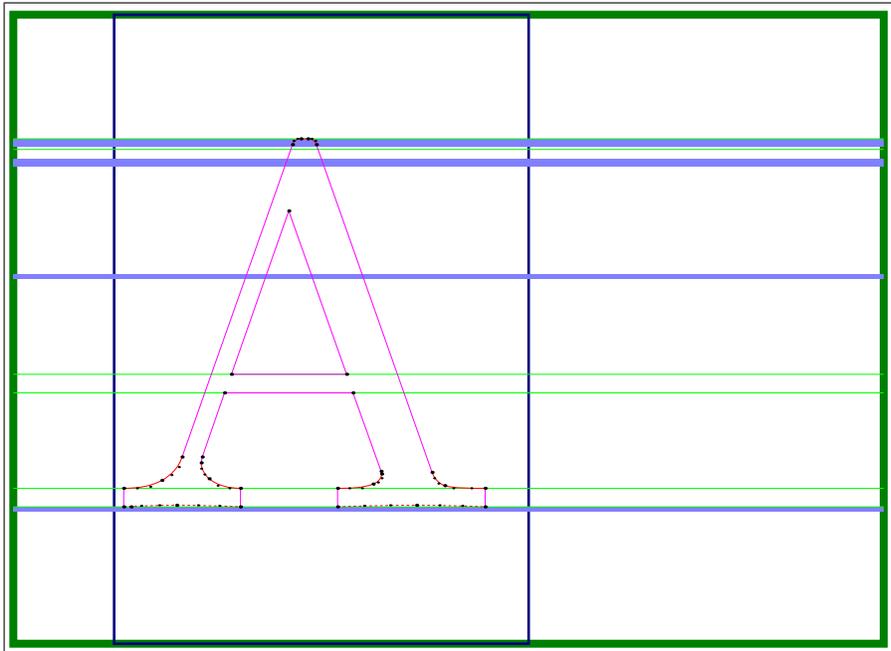
- Fonts are too light to be used with standard printers or in PDF.
- The flex, visible in the “legs” is not properly implemented.

The old MicroPress’ implementation of the EC fonts exhibits similar quality:



It is also darker, to make it more suitable for laser printing and PDF's; but, like the AMS set, it does not have proper flex hinting or blue zones.

Finally, here is the MathPrecise implementation, which is free of all the above problems:

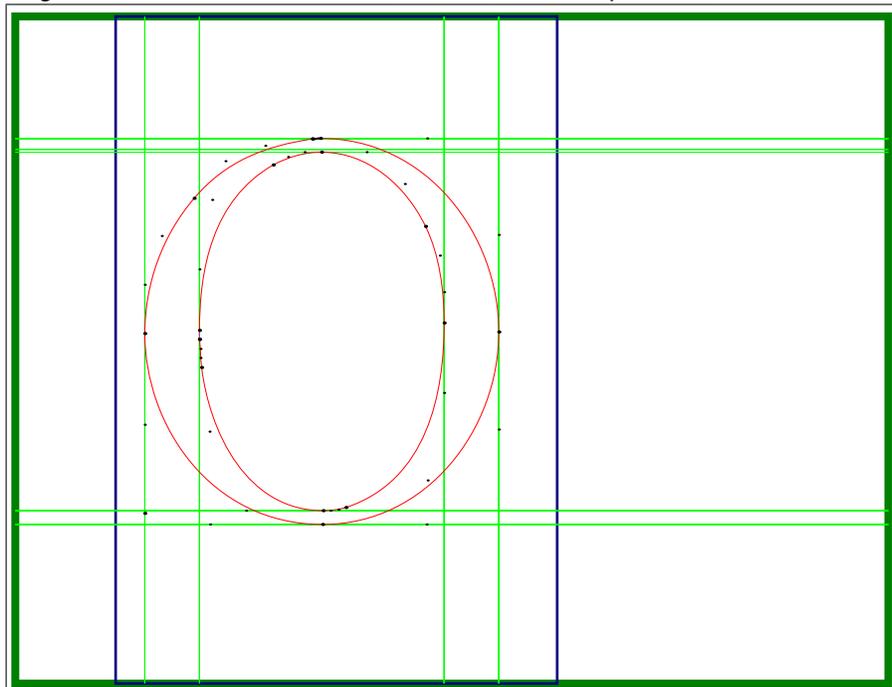


This implementation also includes correct Flex (shown by a dashed line on the picture above) and Blue Zone coding:

```
MAMA MAMA MAMA MAMA MAMA
```

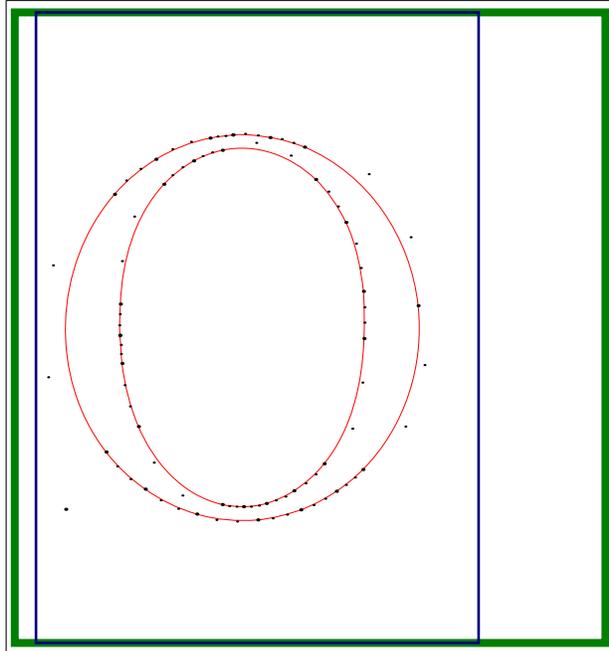
1.2 The Letter O

Turning our attention to the letter “O”, we see, in the SuperFont `\tenrm` font



The two errors correspond to a straight line segment inserted into what is supposed to be a round shape (left side of the letter), and a strange small curve that appears on the top.

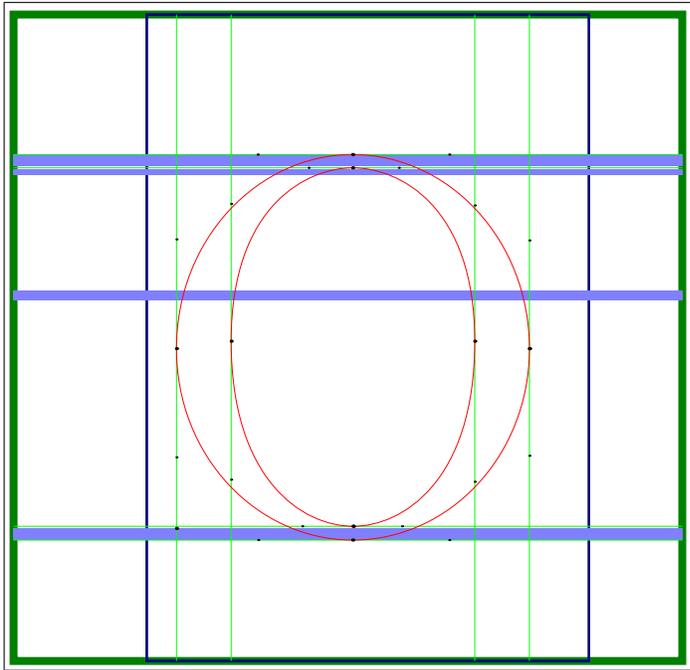
The TEX trace implementation,



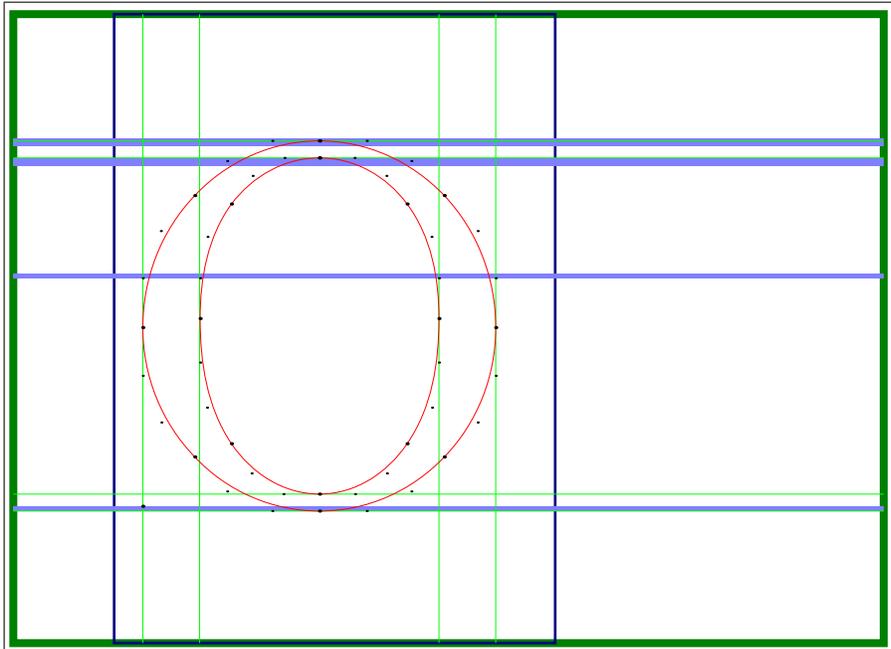
again attempts to build a round shape by using very large number of points; but, amazingly, it neglects to put a point at the left extremum of the letter. Together with the the absense of hinting it makes it probably that that the glyph would have different darkness on the left and right sides.

The other implementations are free from such snafu's.

For comparison, here is the `cmr10` implementation



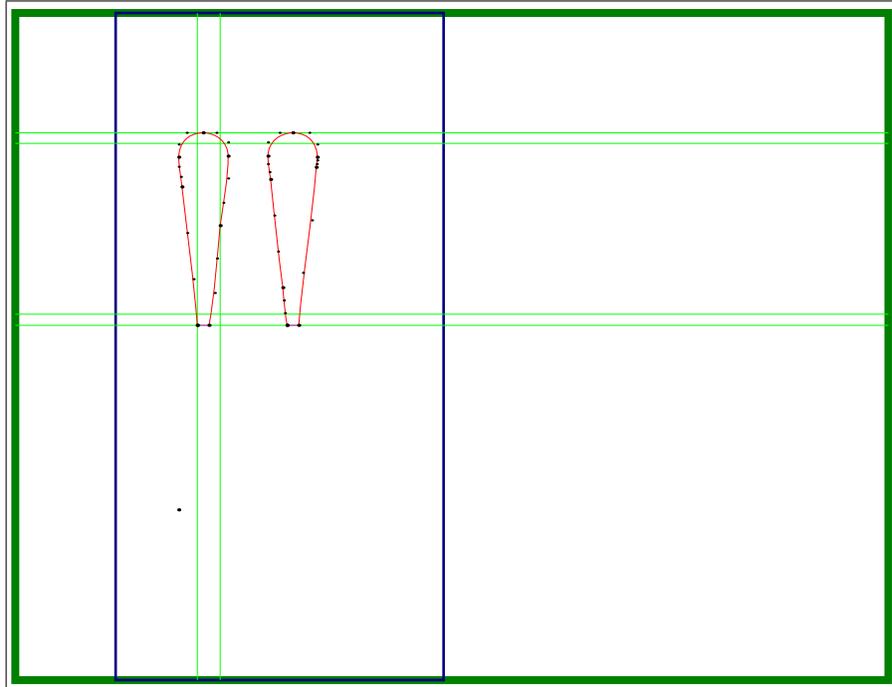
and the MathPrecise implementation:



1.3 The Double Quote

For the final character, we turn to the double quote. This character is not present in the `cmr10` font, so here we only compare the glyph from the SuperFont set against the proper implementation.

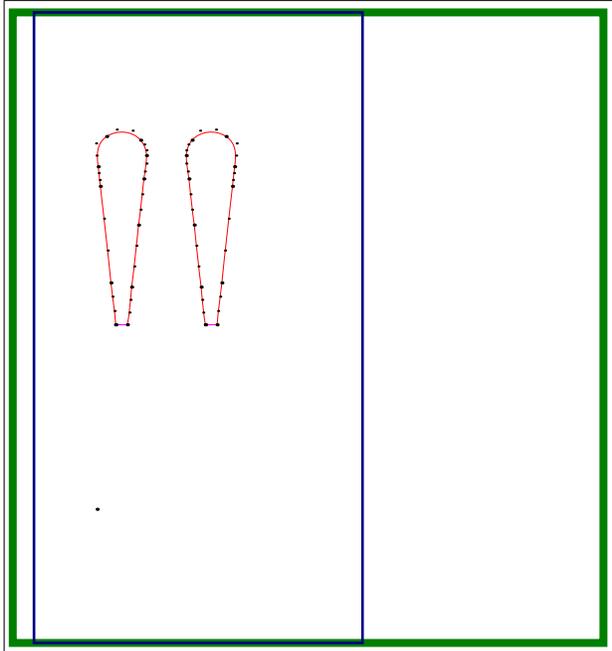
In the SuperFont glyph,



we immediately see one common effect of a badly tuned up autotracer: the two parts of the quote glyph are actually different; loading this glyph into a font editor will actually show different errors in both parts.

For reasons which defeat an easy explanation the left side of the quote is hinted (improperly) but the right one is not.

In the $\text{T}_{\text{E}}\text{X}$ trace version shown heres



we see another common error: missing point at the top of the quotes; correctly coded fonts must have control points at the extremum points:

No such problems in the MathPrecise version:

