Conversion of TeX fonts into Type 1 format

Szabó Péter

<pts@inf.bme.hu>

September 2001
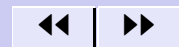
presented at EuroTeX 2001

Home Page

Go Back

Full Screen

Close   Quit

❧ Motivation ☙

The problem: Acrobat Reader renders most fonts slowly and unreadably ugly on screen. This is because most TeX fonts can be included into PDF files only as high resolution raster (bitmap) images, and Acrobat Reader shows such images slowly and inaccurately.

☞ example: nice, with Type 1 font

☞ example: ugly, with bitmap font generated by METAFONT

Solution: convert the offending TeX fonts into something that Acrobat Reader can display well, and make `pdftex` embed the converted fonts into the PDF file. The best candidate for this font format is Adobe Type 1.

◀◀ ▶▶

◀ ▶

Page 2 of 12

Home Page

Go Back

Full Screen

Close | Quit

✂ Bitmap and vector fonts ✂

Bitmap fonts:

- easy and fast to render

- device specific

- resolution specific

- generated, not post-processable

- print nicely iff DPI is large

- separate, small screen version

- examples: TEX .pk, TEX .gf, X11 BDF

Vector outline fonts:

- need complicated rendering

- device independent

- scalable, transformable

- editable

- always print nicely

- hard to read on screen

- examples: Type 1, OpenType, TrueType, TEX .mf

Home Page

Go Back

Full Screen

Close | Quit

✄ Format comparison ✄

Type 1 files (`.pfa`, `.pfb`):

- ✇ vector outline format
- ✇ work with `tex` and `pdftex`
- ✇ appeared with PostScript (1985)
- ✇ supported by most OSs and DTP software
- ✇ 3rd order (cubic) Bézier curves
- ✇ filled regions only
- ✇ embedded into PDF verbatim, the PDF-viewer renders them
- ✇ rendered nicely in Acrobat

METAFONT source files (`.mf`):

- ✇ vector outline format
- ✇ work with `tex` and `pdftex`
- ✇ appeared with TEX (1983)
- ✇ only METAFONT understands them (not even METAPOST)
- ✇ 3rd order (cubic) Bézier curves
- ✇ filled and stroked regions etc.
- ✇ already rendered before embedded to PDF
- ✇ displayed ugly in Acrobat

◀◀ ▶▶

◀ ▶

Page 4 of 12

Home Page

Go Back

Full Screen

Close | Quit

✄ Possible solutions ✄

- ☙ avoid conversion, use existing fonts and glue them together with *virtual fonts* (`.vf`). Example: AE (CM→EC). Drawback: missing glyphs, missing design sizes.

- ☙ *design new* fonts parallelly in both formats. Example: METATYPE1 (at EuroTEX 2001). Drawback: cannot convert existing fonts.

- ☙ *modify* METAFONT to output vector fonts. Drawback: too much effort.

- ☙ *post-process* METAPOST's output. Example: MetaFog. Drawbacks: METAPOST can't understand all `.mf` files, Type 1 requires special contours without intersections etc., needs human intervention.

- ☙ *trace* METAFONT's bitmap output. Example: TEXtrace. Drawback: minor quality loss.

- ☙ a glyphwise *mixture* of these

Home Page

Go Back

Full Screen

Close　Quit

✂ T_EXtrace ✂

T_EXtrace is a collection of scripts for UNIX that convert *any* T_EX font into a Type 1 outline font immediately suitable for use with T_EX. The documents using these fonts cannot be visually distinguished from those using the originals, moreover PDF documents show up quickly and nicely in Acrobat Reader.

http://textrace.sf.net/

The operation of T_EXtrace:

1. calls `dvips` and `gs` to render all the 256 glyphs in high ($\geq 7000\,\text{DPI}$) resolution

2. calls `autotrace` to convert each bitmap to outlines

3. makes syntactical corrections, positions glyphs to their origin and assembles them to a Type 1 `.pfb` file

Home Page

Go Back

Full Screen

Close Quit

� Benefits ✺

- ☙ accepts *any TₑX font* (`.mf`, `.pk`, Type 1, TrueType etc.)

- ☙ operates completely *automatically*, finishes one file in ≈ 20 minutes without asking any questions

- ☙ is *free*, licensed under the GNU GPL

- ☙ produces *portable Type 1 output*, which avoid bugs in many third-party software

◀◀ ▶▶

◀ ▶

Page 7 of 12

Home Page

Go Back

Full Screen

Close   Quit

✂ Screen shot ✂

```
> autotrace tmp/tmp_char.pbm -filter-iterations 9 -background-color FFFFFF
> gs -d_n=252 -dNODISPLAY -q tmp/gstderr.ps trace2.ps
> autotrace tmp/tmp_char.pbm -filter-iterations 9 -background-color FFFFFF
> gs -d_n=253 -dNODISPLAY -q tmp/gstderr.ps trace2.ps
> autotrace tmp/tmp_char.pbm -filter-iterations 9 -background-color FFFFFF
> gs -d_n=254 -dNODISPLAY -q tmp/gstderr.ps trace2.ps
> autotrace tmp/tmp_char.pbm -filter-iterations 9 -background-color FFFFFF
> gs -d_n=255 -dNODISPLAY -q tmp/gstderr.ps trace2.ps
> perl ./t1d2gsx.pl --256
255 CharBBox entries.
> perl -x type1fix.pl tmp/tmp_gen1.gsx pfb: gcti1000.pfb --pack --dump-spaces=no
--debug-warnings --chk-insize=no
infile tmp/tmp_gen1.gsx
insize 370868
will preserve comments
ok internal interp
ok internal poss
packed 257 chars
writing PFB
outfile gcti1000.pfb
outsize 101784
check: OK /tmp/_t1f_2936_1: /TeX-ecti1000
done
~$ _
```

Home Page

Go Back

Full Screen

Close   Quit

❧ Problems remaining ❧

- huge font files ($3.15 \times$)

- no hinting in output

- AutoTrace mis-recongnizes some corners

- METAFONT fails for some fonts

- limited portability (needs UNIX, teTeX, `bash`, `perl`, `gs`)

- written for experts

- no Unicode support, limited to 256 characters

- doesn't convert metrics (but original `.tfm` is OK for TeX)

◀◀ ▶▶

◀ ▶

Page 9 of 12

Home Page

Go Back

Full Screen

Close Quit

� Quality comparison ✂

Original METAFONT fonts:

- ⚬ printed perfectly ($\geq 600\,\text{DPI}$)
- ⚬ displayed nicely in `xdvi`
- ⚬ displayed awful in `gs`
- ⚬ displayed tolerably in `gv -an`
- ⚬ displayed awful in `acroread`

Fonts converted by TEXtrace:

- ⚬ printed nicely ($\geq 300\,\text{DPI}$)
- ⚬ displayed nicely in `xdvi`
- ⚬ displayed awful in `gs`
- ⚬ displayed nicely in `gv -an`
- ⚬ displayed nicely in `acroread`

Reasons:

- ⚬ Acrobat Reader hates bitmap fonts
- ⚬ screen resolution ($\leq 100\,\text{DPI}$) is too small, but antialiasing solves the problem
- ⚬ hinting without antialiasing is not enough for Type 1 or METAFONT.

⚡ The ideal solution ⚡

1. *merge* METAFONT and METAPOST

   - allow both bitmap a PostScript output
   - allow elliptical pens, METAFONT pictures etc.
   - support all METAFONT language dirty tricks

2. *post-process* the PostScript output (better than MetaFog)

   - convert strokes (etc.) to fills
   - remove contour overlaps
   - reorganize touching contours
   - do other small modifications for Type 1 compliance

3. *guess* most *hinting* information automatically (really hard)

4. implement an *effective human interface* to modify hinting

5. convert metrics, kerning, ligatures, support Unicode

Of course, none of the steps have been implemented yet.

Home Page

Go Back

Full Screen

Close | Quit

✂ Conclusion ✂

⚭ T_EXtrace already works, is automatic and free

⚭ T_EXtrace can be used by anyone who understands the problem, but can't solve it by him/herself

⚭ professional quality requires a font expert (preferably a font designer) much time.

⚭ the ideal, almost automatic solution requires too much resources to implement

⚭ Not even mayor T_EX fonts are expected to be freely available in high quality Type 1 for years. Until this happens, T_EXtrace is the best alternative.

◀◀ ▶▶

◀ ▶

Page 12 of 12

Home Page

Go Back

Full Screen

Close Quit