

Hanging punctuation

a pdf \TeX microtypographic extension

Hans Hagen
Ridderstraat 27, 8061GH
Hasselt
pragma@wxwxs.nl

keywords

micro typography, margins, kerning, pdf \TeX , Con \TeX t

abstract

At the time of this writing, Hàn Thế Thành is writing his thesis, wherein he explains in great detail two extensions to \TeX 's typesetting engine. In this article we will explore marginal kerning, or character protruding. We will demonstrate the effects and introduce the new primitives, as well as explain how to enable this feature in Con \TeX t.

Introduction

When breaking a paragraph into lines, only a few typographic engines consider the paragraph as unit of information and \TeX is one of them. Although \TeX has a pretty good reputation when it comes to breaking paragraphs, it is no secret that micro-typographic improvements are not easily achieved.

After some years of experimenting, as part of his promotion, Hàn Thế Thành, the author of pdf \TeX , has extended \TeX 's core line breaking algorithm with two micro-typographic features: *hz* optimization and marginal kerning. As soon as you know that the acronym *hz* represents the name of a typographer, the name of Hermann Zapf will spring into mind. In the spirit of Gutenberg, he has invented a method for visually improving the grayness of a paragraph by slightly changing the shape of a glyph.

A maybe less impressive, but certainly worthwhile second extension to the \TeX typesetting engine is marginal kerning, also called protruding. Where *hz* optimization demands additional font resources, and thereby is more complicated to support, protruding can be supported with minimal effort from the user. Therefore, in this article, we will focus on protruding and leave *hz* for another article, if only because it would put some demands on the production process of the MAPS.

Protruding characters

In the following fake paragraph, you can see a hyphenation point, a secondary sentence, separated by a comma, and a last sentence, ending with a period. Miraculously, this paragraph fits into lines. Although exaggerated, these lines demonstrate that visually the hyphen and punctuation characters make the margin look ragged.



Before computers started to take over the traditional typesetter, it was common practice to move hyphens and punctuation into the margin, like in:





In this alternative, the overall appearance is more gray, and therefore enhances the process of reading.

Sometimes, shifting the characters completely into the margin is too much for the sensitive eye, for instance with a slanted font, where the characters already hang to the right. In such cases, we need to compromise.



In PDF_T_E_X protruding is implemented in two ways. The first method does not take protruding into account when breaking the paragraph, but the second method does. In method one, the characters are moved into the margins after the paragraph is broken, and this move influences the inter-word spacing. As said, the second method takes the shifts into account when breaking the paragraph into lines. This means that when method two is enabled, the line-breaks may differ from the normal situation.

Since we want optimal typesetting, we will focus on protruding method two. For this purpose we will use a quote from Hermann Zapf's article "About micro-typography and the *hz*-program" in *Electronic Publishing*, vol 6 (3), 1993.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

After $T_E X$ has typeset this paragraph, it has constructed the following lines.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software...There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.....

As you can see, the height and depth of the lines depend on the characters, but their width equals what $T_E X$ calls `\hspace`. However, the natural width of the lines may differ from `\hspace`.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

Here the inter-word space is fixed to what T_EX considers to be a space. This example also demonstrates that T_EX does not have spaces, but stretches the white area between words to suit its demands. When breaking lines, T_EX's mind is occupied by boxes, glue and penalties, or in more common language: (parts of) words, stretchable white space, and more or less preferred breakpoints.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the in-	struction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design.	Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.
---	---	---

This time we have enabled PDF_TE_X's protruding mechanism. The characters that stick into the margin are taken into account when breaking the paragraph into lines, but in the final result, they do not count in the width. Here we used an ugly three column layout so that we got a few more hyphens to illustrate the principle, but in the next examples we will stick to two columns.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic	instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.
--	--

In this first example we just typeset the text in the traditional way. The hyphens and punctuation fit into the margin.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic	instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.
--	--

In this example, the protruding machinery is put to labour. The hyphens and punctuation may now stick into the margin completely. The next two examples shows what happens when we limit the protruding to 75% and 50% respectively.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction	manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good
---	---

and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic

the screen, will make everything automatic from now on.

instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

Although available in PDF \TeX , this feature is not limited to PDF output. Also, when protruding is not enabled, or when level one protruding is used, the output is 100% identical.

The new primitives

The way protruding is implemented is, as you may expect from a \TeX extension, rather advanced. Although such features make most sense when encapsulated in the font switching mechanism, we will now introduce the new primitives. First we define a font, say a Times Roman, using \TeX 's `\font` primitive.

```
\font\NiceFont=tir at 12pt
```

For this font, we want to stick the hyphen and some punctuation into the right margin. The `\rprcode` primitive (globally) assigns a factor to a specific character slot in a font. The `\-` is \TeX 's way of calling up the character number associated with the hyphen.

```
\rprcode\NiceFont\ -=700
\rprcode\NiceFont\ .=700
\rprcode\NiceFont\ \,=700
\rprcode\NiceFont\ \:=700
\rprcode\NiceFont\ \;=700
```

A factor of 700 means that the hyphen will move 70% of its width into the (here) right margin. You can specify the left protruding factor in a similar way using the `\lprcode` primitive.

These definitions demonstrate that each character can be set up to protrude, so we can say:

```
\rprcode\NiceFont\ !=200
```

This will move the exclamation mark some 20% into the right margin when it ends up there. In this way, for each font dedicated shifts can be specified. Although hanging punctuation can be faked with macros to a certain extent, it will be clear that the fact that you have control over each character, combined with tight integration in the paragraph builder, beats every macro based solution.

So far, we have only defined protruding factors. The protruding mechanism itself is enabled with the low level primitive `\pdfprotrudechars`. When set to 1, you get level one protruding, which does not influence the line breaks, but for optimal results, you should enable level two protruding.

Support in Con \TeX T

At this moment, Con \TeX T supports protruding as an experimental extension to its font switching mechanism. Since protruding is related to the shape of the font, setting it up (currently) takes place in a similar way as encodings.

```
\definefontsynonym [Times-Roman] [tir] [encoding=ec,handling=normal]
```

Of course it would be best to make protruding specification part of the font specification, You can read more about this in Hàn Thế Thành's thesis, where he also reports the opinions of Donald Knuth and Hermann Zapf on his extensions as well as some alternative approaches.

Defining a protruding vector in `CONTEX`T does not differ much from other encoding specific definitions. The default punctuation vector looks like:

```
\startfonthandling[defpunc]
  \defineprotrudefactor ! 0 .2
  \defineprotrudefactor , 0 .7
  \defineprotrudefactor - 0 .7
  \defineprotrudefactor . 0 .7
  \defineprotrudefactor : 0 .5
  \defineprotrudefactor ; 0 .5
  .....
\stopfonthandling
```

Instead of using numbers (like the 700 in the previous definition), we use fractions, one for the left and one for the right shift. There are a few more vectors defined, like `defalph` for characters, where we only apply very small shifts.

A combination of such vectors is packaged in a font handler using the following command:

```
\definefonthandling [normal] [defpunc,defalph]
```

Since these definitions depend on the encoding, they are loaded with the `\useencoding` command, for instance:

```
\useencoding[pro]
```

Again, in `CONTEX`T we don't use the low level primitive `\pdfprotrudechars` directly, but we have integrated it into the normal alignment macros. Apart from other advantages, this means that we can guarantee compatibility with other `TEX`'s.

```
\setupalign[hanging] % in dutch: \steluitlijnenin[hangend]
```

Font protruding is turned off automatically in controlled situations, and more control will be added in due time. We already mentioned that the amount of protruding depends on the shape, which is why we may need different values for slanted and bold shapes. One way of using the same vector for different shapes, is changing the strength of the protruding:

```
\definefonthandling [slanted] [defpunc] [right=1.5]
```

Here, because a slanted glyph already sticks into the margin, we limit protruding to punctuation.

It will take some time and experiments to define suitable vectors for different classes of shapes and determine the optimal values for the strength. But, if this feature is found useful, you may count on `CONTEX`T to come with a suitable set of such vectors. In spite of what you may expect from the previous examples, the user can limit his or her effort to enabling this nice feature by `\setupalign`.