letterspaceadjust

v0.0b
Arno Trautmann
arno.trautmann@gmx.de

Abstract

The package letterspaceadjust offers a simple way to increase the typographic quality of very narrow columns. This is achieved by inserting stretchable space between all glyphs, so the text can be spaced out more evenly. The result is *not* very nice, but maybe the best to achieve in narrow columns.

This is the documentation of the package `letterspaceadjust`. The package provides an experimental approach of improving the result of typesetting in very narrow columns. For this, the distance between letters is stretched evenly by a very small amount – in the best case so small that you hardly notice. In the worst case so much that it disturbs the reading. This very implementation is just for testing purposes – please play around with the parameters and report your results so I can see whether this package will be of any use. If so, an official CTAN release will follow when the package is stable and useable.

Attention: This package is under development and everything presented here might and will be subject to incompatible changes.

If you have any suggestions or comments, just drop me a mail, I'll be happy to get any response! The latest source code is hosted on github – Feel free to comment or report bugs there, to fork, pull, etc.: `https://github.com/alt/letterspaceadjust`

Contents

I User Documentation 3

1 How do I use it? 3

1.1 Controlling the parameters . . . 3

1.2 Example use . . . 4

II Implementation 5

2 The LaTeX package: letterspaceadjust.sty 5

3 Lua Module 6

3.1 setup of variables . . . 6

3.2 function implementation . . . 7

3.3 textletterspaceadjust . . . 7

Part I
User Documentation

1 How do I use it?

The basic macro of this package is `\letterspaceadjust`. Using this macro, one activates the additional spacing for the rest of the document – until an `\unletterspaceadjust` stops it. These macros work per paragraph, so you cannot (de)activate it for single lines. For this, you can use the macro `\textletterspaceadjust` or `\textlsa{}` which activates spacing for everything inside the braces.

There is one feature I'm not sure about so far: Ligatures. For large interglyph spacing, it doesn't look right to let ligatures together. However, breaking them looks bad in the case of small or no additional spacing. Now, that's your job: Test what looks best for you and report it to me. This package will support both versions, however I wonder what the default value should be. For now, it is breaking of ligatures which will start when calling `\letterspaceadjust` and stop by `\unletterspaceadjust`. Everything outside will be ligatured normally. For all this, fontspe (actually luaotfload) is assumed as ligaturing there happens at another place than is normally thought for. (In the `pre_linebreak_filter` instead of the `ligaturing` callback).

1.1 Controlling the parameters

Using letterspaceadjust without the microtype package is dumb. Therefore, the package requires microtype and offers two commands for convenience to change the microtype parameters. These are just shorthands and shall help in finding good parameters:

`\lsamicrostretch{}` Changes the amount by which glyphs are extended. Default value in microtype is 20 (measured in 1/1000, so 20 means glyphs are extended at max. 2% of their width) However, I find that values up to 40 are useable and improve the result in narrow columns. Try it out. A larger expansion also helps to increase the greyness which is needed when we increase the letter spacing.

`\lsamicroshrink{}` Same for the amount of shrinking. By default, this is also set to 20 and if not changed explicitely, the shrink value is the same as the stretch value. I think that the shrinking should not be increased when using letterspaceadjust, so try setting (i. e. keeping) it to 20.

The second way of maybe improving the quality of very narrow columns is to increase the value of `\emergencystretch`. In simple words, this is the amount of additional stretchable material that is added if TeX cannot find a good solution of the paragraph. This may lead to very loose columns, but that is exactly what the letterspacing will improve. So a larger emergencystretch will help us here. Maybe. If you really want to understand what `\emergencystretch` does, please consult the TeXbook.

`\lsaemergency{}` Sets the additional stretchable material. Try values below `1em` or so. Use font related units (as `em`) to be independent of the current font.

Finally, there are macros that directly change what this package does. Mainly this is the amount of spacing that is used between the glyphs. This value should be very small; everything above `0.05em` looks too loose and unreadable.

`\lsastretch{}` Sets the amount of stretch to be inserted between the glyphs. Values below `0.05em` seem reasonable.

`\lsakeepligatures` Keep the ligatures as they are. May look bad with large interglyph spacing.

In a final version of this package, there will be of course a key-value interface and also a set of predefined values à la "very strong letterspacing", "light letterspacing" etc. Do not forget to consider the `\textletterspaceadjust`-variant to be used when the linewidth changes during the paragraph. (e. g. pictures in the text)

1.2 Example use

For testing, you might want to consider the following example. Remember that you cannot change the amount of microtypes stretching after `\begin{document}`, therefore I cannot show different expansions here.

The examples are typeset with the following setting, from left to right. The amount of font expansion in this document is set more than twice as high as the default value (45 instead of 20) to show more effect.

- without expansion or letterspacing,
- with expansion, but without letterspacing
- without expansion, but with letterspacing
- with expansion and letterspacing

This is a small text with some words that should show how to use this package.

This is a small text with some words that should show how to use this package.

This is a small text with some words that should show how to use this package.

This is a small text with some words that should show how to use this package.

The default values correspond to the following settings:

```
\lsaemergency{0.0em}
\lsastretch{0.02em}
\lsamicrostretch{20}
\lsamicroshrink{20}
```

Part II
Implementation

2 The LaTeX package: letterspaceadjust.sty

Everything to get stuff working from the TeX side. Here, only a .sty file is provided and plain/ConTeXt users have to find their way. I'll happily support them, though!

The Lua file is not found by using a simple `dofile("letterspaceadjust.lua")` call, but we have to use kpse's `find_file`.

```
\ProvidesPackage{letterspaceadjust}%
  [2013/03/10 v0.0b letterspaceadjust package]
\input{luatexbase.sty}
\RequirePackage{luacode}
\directlua{dofile(kpse.find_file("letterspaceadjust.lua"))}
```

Commands for (un-)registering the functions in the callbacks.

```
\def\letterspaceadjust{
  \directlua{
    if (lsakeepligatures == false) then
      luatexbase.remove_from_callback("pre_linebreak_filter","luaotfload.pre_linebreak_filter")
    end
    luatexbase.add_to_callback("pre_linebreak_filter",letterspaceadjust,"letterspaceadjust")
  }
}
\def\unletterspaceadjust{
  \directlua{
    luatexbase.remove_from_callback("pre_linebreak_filter","letterspaceadjust")
    if (lsakeepligatures == false) then
      luatexbase.add_to_callback("pre_linebreak_filter",nodes.simple_font_handler,"luaotfload.pre_
    end
  }
}
```

Synonyms, just to be funny:

```
\let\stealsheep\letterspaceadjust     %% synonym in honor of Paul
\let\unstealsheep\unletterspaceadjust
\let\returnsheep\unletterspaceadjust
```

Now the setup for the `\text`-versions. We utilize LuaTeXs attributes to mark all nodes that should be manipulated. The macros should be `\long` to allow arbitrary input.

```
\newluatexattribute\letterspaceadjustattr

\long\def\textletterspaceadjust#1{
  \setluatexattribute\letterspaceadjustattr{42}#1\unsetluatexattribute\letterspaceadjustattr
  \directlua{
    if (textletterspaceadjustactive) then else % -- if already active, do nothing
```

```
      luatexbase.add_to_callback(
        "pre_linebreak_filter",textletterspaceadjust,"textletterspaceadjust")
    end
    textletterspaceadjustactive = true         % -- set to active
  }
}
\let\textlsa\textletterspaceadjust
```

Macros to control the behaviour of the package. First, macros that go to the microtype package:

```
\def\lsamicrostretch#1{\microtypesetup{stretch=#1}}
\def\lsamicroshrink#1{\microtypesetup{shrink=#1}}
```

Then a macro that just executes `\emergencystretch=#1`

```
\def\lsaemergency#1{\emergencystretch=#1}
```

Finlly, the parameters of the lua function, mainly the amount of stretchability.

```
\def\lsastretch#1{%
  \directlua{
    lsa_stretch_ind = lsa_stretch_ind + 1
    letterspace_spec[lsa_stretch_ind] = node.copy(letterspace_spec[1])
    letterspace_glue.spec    = letterspace_spec[lsa_stretch_ind]
    letterspace_spec[lsa_stretch_ind].stretch = tex.sp"#1"
  }%
}

%
```

3 Lua Module

This file contains all the necessary functions and is the actual work horse of this package. Yet another piece of code by Paul. This is primarily inteded for very narrow columns, but may also increase the overall quality of typesetting. Basically, it does nothing else than adding expandable space *between* letters. This way, the amount of stretching between words can be reduced which will, hopefully, result in the greyness to be more equally distributed over the page.

Why the synonym `stealsheep`? Because of a comment of Paul on the `texhax` mailing list: http://tug.org/pipermail/texhax/2011-October/018374.html

3.1 setup of variables

First, we set up some constants that are used by many of the following functions. These are made global so the code can be manipulated at the document level, too.

```

lsakeepligatures = false

local nodenew = node.new
local nodecopy = node.copy
local nodeinsertbefore = node.insert_before
local nodeinsertafter = node.insert_after
local nodeid = node.id
local nodetraverseid = node.traverse_id

Hhead = nodeid("hhead")
RULE = nodeid("rule")
GLUE = nodeid("glue")
WHAT = nodeid("whatsit")
COL = node.subtype("pdf_colorstack")
GLYPH = nodeid("glyph")

lsa_stretch_ind = 1
letterspace_spec = {}
letterspace_spec[lsa_stretch_ind] = nodenew(nodeid"glue_spec")
letterspace_glue = nodenew(nodeid"glue")
local letterspace_pen = nodenew(nodeid"penalty")

letterspace_spec.width   = tex.sp"0pt"
letterspace_spec[lsa_stretch_ind].stretch = tex.sp"0.02em"
letterspace_glue.spec    = letterspace_spec[lsa_stretch_ind]
letterspace_pen.penalty  = 10000
```

3.2 function implementation

```
letterspaceadjust = function(head)
  for glyph in nodetraverseid(nodeid"glyph", head) do
    if glyph.prev and (glyph.prev.id == nodeid"glyph" or glyph.prev.id == nodeid"disc" or glyph.pr
      local g = nodecopy(letterspace_glue)
      nodeinsertbefore(head, glyph, g)
      nodeinsertbefore(head, g, nodecopy(letterspace_pen))
    end
  end
  return head
end
```

3.3 textletterspaceadjust

The `\text...`-version of `letterspaceadjust`. Just works, without the need to call `\letterspaceadjust` globally or anything else. Just put the `\textletterspaceadjust` around the part of text you want the function to work on. Might have problems with surrounding spacing, take care!

```
textletterspaceadjust = function(head)
  for glyph in nodetraverseid(nodeid"glyph", head) do
    if node.has_attribute(glyph,luatexbase.attributes.letterspaceadjustattr) then
      if glyph.prev and (glyph.prev.id == node.id"glyph" or glyph.prev.id == node.id"disc") then
        local g = node.copy(letterspace_glue)
        nodeinsertbefore(head, glyph, g)
        nodeinsertbefore(head, g, nodecopy(letterspace_pen))
      end
    end
  end
  luatexbase.remove_from_callback("pre_linebreak_filter","textletterspaceadjust")
  return head
end
```