BubbleKern

Developed by Toshi Omagari

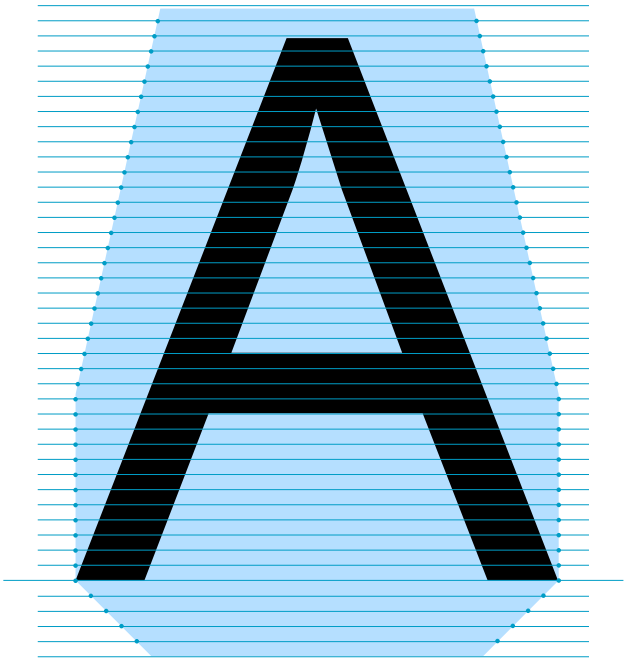
Manual released on 1 September 2015

What is BubbleKern

BubbleKern is a set of new kerning tool that is based on the shape of surrounding outlines you draw, which is called a *bubble*. Once you draw bubbles for all letters, BubbleKern automatically kerns the typeface in seconds. That's right, you kern by drawing it!

A bubble needs to be made as a extra layer in each master, and it has to be named "bubble". It doesn't have to be straight segments or single path, and you craw however you want. Components automatically inherit the bubbles of base glyphs, so you don't have to draw one for an accented glyph (for letters like Eth or Tbar, you just draw a bubble for the bar).

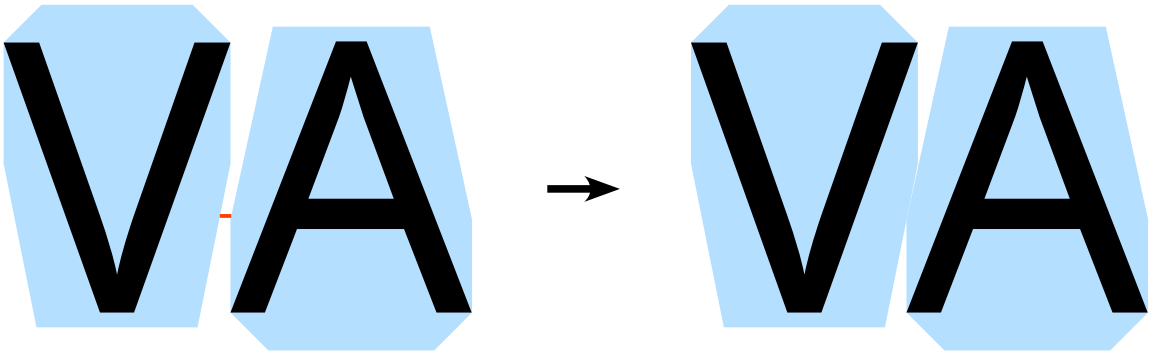
When asked to kern, first of all BubbleKern scans through each letter to make an approximation of bubble shape so that it's faster to calculate. And the temporary dictionary of bubble sidebearings is built for every letter.



Scan step is every 20 units starting from 0. 20 may seem rough and I may change that in the future, but the result has been pretty good so far. More accurate step values like 1 are meaninglessly so.

Update: it's every 10 units now. A bit slower, but a bit more accurate.

Then for a given pair, BubbleKern looks at every scan height, finds where bubbles touch each other first, and kerns by the distance. In the example below, it finds the closest distance of 30 units, and kerns by -30.



As for pairs to kern, a type designer needs to make a pair list for BubbleKern to work with. Don't worry, there is a built-in pair builder for that.

But why does it not simply look at all possible pairs, every glyph against every other glyph? Because that takes so much time, only to create a huge mess. If your font has 600 glyphs, the maximum number of possible pairs is 600^2=360,000 which contains meaningless pairs like %& or some other junk that you won't enjoy cleaning up later.

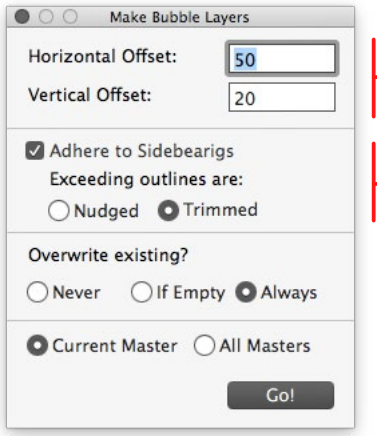
I believe that a good tool is something the user can control. BubbleKern will kern a font for you, but only do the obvious, not do some mysterious calculation in the background (e.g. Adobe's optical kerning). That's why I leave bubble shapes and pairs for the designer to define.

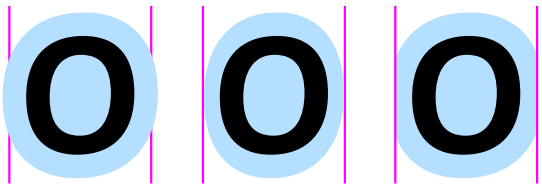
Tools

1. Make Bubble Layers

A Python script to help you start drawing bubble layers. Without this, you would need to make a bubble layer by hand. You have a a variety of options to get a bubble shape you want. I personally recommend using *Adhering to Sidebearings* option for most glyphs, because making bubbles stick to sidebearings is the easiest way to avoid unnecessary kerning. A bubble that ignores sidebearings will be kerned everywhere, unless you compensate for it in other glyphs.

Same as offset filter





Ignored Nudged Trimmed

It may be good to ignore sidebearings when you are making bubbles for diacritics.

Also, Nudge option does not work well when italic angle is set.

2. Delete Bubble Layers

A Python script that deletes bubble layers from selected glyphs. It only deletes the ones that's in the selected master. No graphical user interface.

3. Show Bubbles

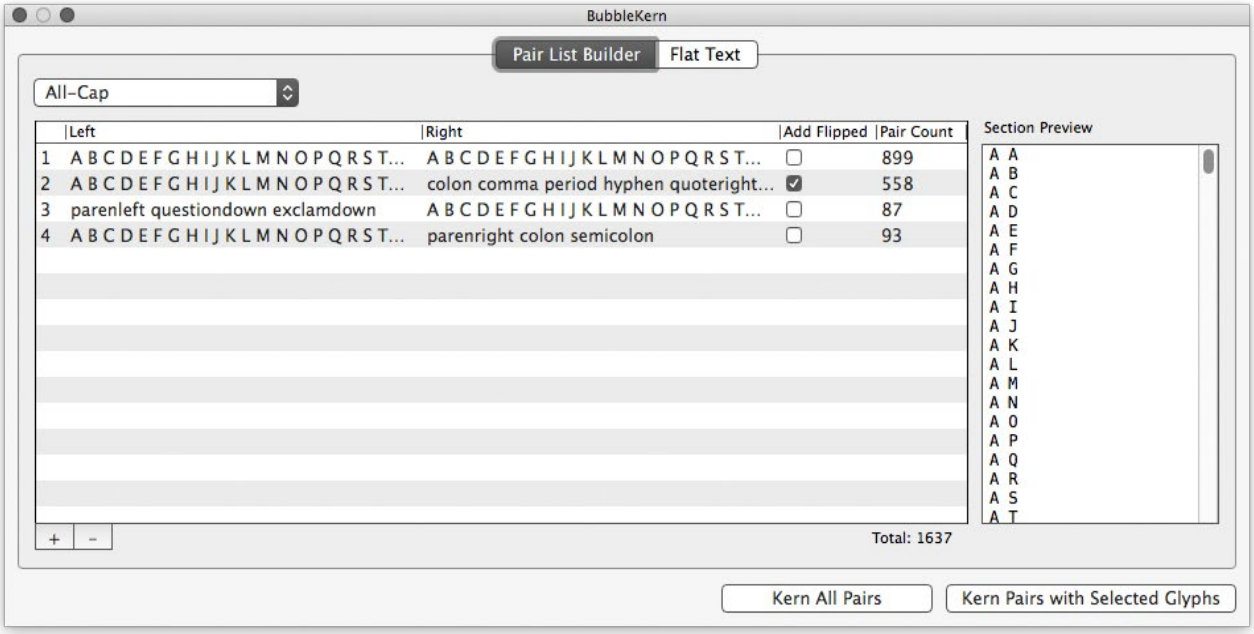
A viewer plugin that shows bubbles in colour. This may become a tool in the future, since you still need to select a bubble layer if you want to edit it. Nevertheless, it's super handy.

4. BubbleKern

The Python script that does the magic. As mentioned, you need to give it a pair list in a form of glyph name against another glyph name. While you can paste a text (it accepts Metrics Machine pair list file), pair list builder is a smart way, where you make a list of permutations. BubbleKern will ignore pairs that; 1. are checked already (duplicate), 2. consist of glyphs that don't exist in the font, 3. consist of glyphs that don't have bubble layers, 4. not related to selected glyphs if you clicked *Kern Pairs with Selected Glyphs*.

I think Pair List Builder is a better way to create a kerning list, but BubbleKern also accepts pair list in plain text, in the same format as MetricMachines's pair list.

The pair list is built as a permutation of Left and Right. You can add flipped pairs except when L and R are the same, in which case you're only creating duplicates. A set of permutations can be saved in Favourites.



Goes through all pairs in the list.

Useful when you are working on one glyph and only want to update pairs related to it, because it's much faster.

Installation

Install three scripts in ~/Library/Application Support/Glyphs/Scripts folder. *Make Bubble Layers* and *BubbleKern* require vanilla and robofab modules to display user interface. You can install them easily from Glyphs (Preferences > Addons > Modules). If you needed to install modules, you will need to restart Glyphs. Installed scripts will appear under *Scripts* menu.

The viewer plugin needs to be installed in ~/Library/Application Support/Glyphs/Plugins folder (if there is none, create one). After installation, you need to restart Glyphs. It will show up under *View* menu as *Show Kern Bubbles*.

If the script gives me an error, you may have glyphsapp.py or .pyc in your Scripts folder. In that case, delete them and try running the script again. Restarting Glyphs may be necessary.

Use

1. Make a bubble layer for each glyph. The layer name should be “bubble” specifically. If you want to create bubble layes for selected glyphs at once, use *Make Bubble Layers...* script. As mentioned in the Tools section, it is important to make the bubble stick to sidebearings somewhere, in order to avoid creating unnecessary kerning (e.g. think about how to avoid creating a kerning value between H and H while using bubbles).

2. Shape your bubble layer, with the help of *Show Kern Bubbles* viewer plugin.

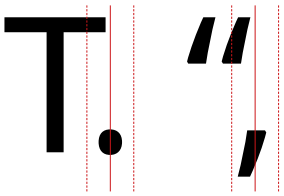
3. After you have bubble layers, run *BubbleKern...*, build your pair list, then click *Kern All Pairs* button. If you are not happy with the result and modify a bubble shape, it is faster to kern only the pairs that are related to that glyph, by clicking *Kern Pairs with Selected Glyphs*. Basically, *All Pairs* is for initial run, and *Selected Glyphs* is for updating specific portion of the kerning data.

Questions & Answers

Is it really faster than the traditional kerning?
Does it actually make a good result?
What is the best bubble shape?
These are essentially the same question, and I don't know the answer yet. I only know that it's more fun to kern with bubbles, and you want to try it.

Does it support group kerning?
No, it's all done on glyph-to-glyph basis, and you need to compress kerning later.

What happens if bubbles may not meet each other in pairs like period and quote mark?
BubbleKern automatically sets a value, which is half the width of the narrower glyph. For example, when you type a string T.T you want to avoid T to touch each other, which will happen if you kern T and period too tightly; text cursor will also appear to go in reverse direction when you navigate by arrow keys. For these reasons, it's safer not to kern more than half of the width of period. At least I think that is the best automatic solution.



Half width of narrower glyph is the safest maximum kerning value to calculate automatically (and easily).

In these two cases where kerning might be too deep or technically infinite, kerning value will be set as half of glyph width of period and comma, respectively.

Kerning values doesn't look very precise. Why?
One reason is efficiency. Because the script approximates the bubble shape, you sometimes might get bubbles overlapping or leaving a little gap. Another time where you see it is that, when the distance between bubbles is too small (less than 8 units), it will be simply ignored. If the distance is 8 units or 9 units, the script will apply a rounded kerning value of 10 (i.e. the smallest kerning value it will apply is 10). Any value larger than 10 will be kerned exactly.

Do you recommend BubbleKern to everyone?
Probably not to an initiate designer. I'd recommend getting used to the old way first, understanding the principle of kerning in digital typography. Although BubbleKern seems closer to what letter spacing should work in principle, it is a more complex tool in reality because it is not easy to draw a bubble that works universally and only generates necessary pairs.

Do you think you can port it to other font editor?
No, for two reasons. Because BubbleKern is a tool that lets you kern a typeface by drawing kerns directly, a font editor needs an interface wherein drawing and spacing is done in the same view, and Glyphs is the only font editor that satisfy the requirement so far. What if you want to use BubbleKern in UFO file? Well, only if there were an editor that works like Glyphs and edits UFO just fine… oh wait, that's Glyphs! Second reason is, I am incompetent at coding for other font editors.