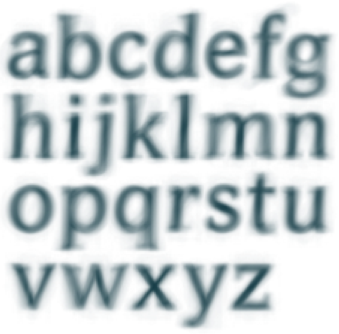
Avería
The *Average* Font

I AM NOT A TYPE DESIGNER. THIS IS THE STORY OF THE CREATION of a new font, Avería: the average of all the fonts on my computer. The field of typography has long fascinated me, and I love playing with creative programming ideas, so it was perhaps inevitable that the idea came to me one day of “generative typography”. A **Google** on the subject brought up little, and I put the idea to the back of my mind until it occurred to me that perhaps the process of averaging, or interpolating, existing fonts might bring up interesting results. Luckily at this point I didn't do any more web searching – instead I grabbed my laptop and came up with an initial idea for finding what the average of all my fonts might look like – by overlaying each letter at low opacity. The results can be seen in the below image.



This was done by printing each letter of each font, at the same point size, to lots of separate images, and then averaging them – using **ImageMagick** and **PHP**. The letters were aligned to the same centre point. I later realised that each font has a **‘baseline’** defined, and an **origin** on that baseline which each glyph is drawn relative to. The same process, repeated with equal origins, gives slightly different results (see below) – here you can see the baseline is very well-defined, with the glyphs becoming more blurred

Yclept
Planchant
Ribald pulchritude
Honorificabilitudinitatibus

— · —

Quincunx
LATCHSTRING
Unguent lascivious
Cuneiform canoodling spandrels

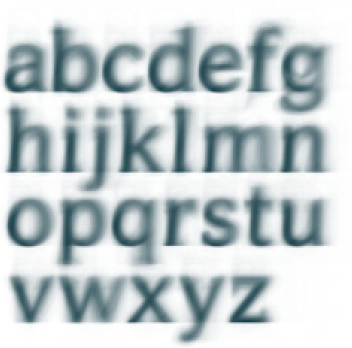
— · —

Spatchcock
Piñata jalapeño
Frappé (naïve) soufflé
Bête noire; blasé château
Cause célèbre: continuüm. Coup d'état.

— · —

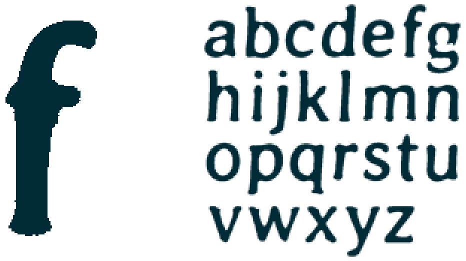
Smörgåsbord
Soupçon mêlée
Götterdämmerung
Crème brûlée débâcle

towards the top right of each.

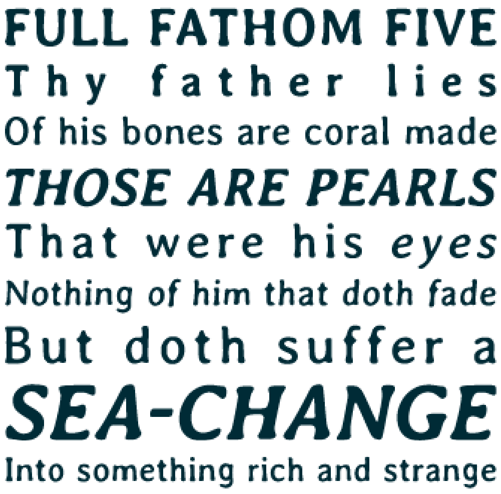


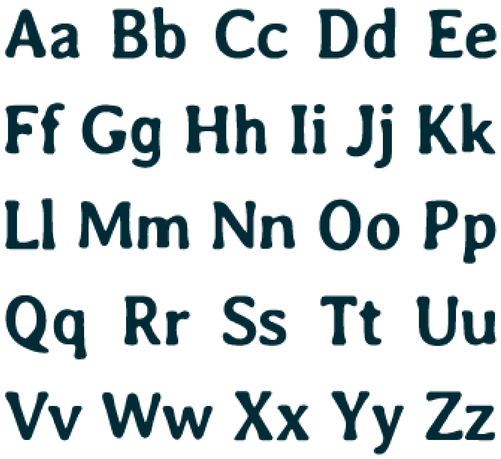
I was quite pleased with the results. It was only later that I discovered **this had already been done** – though it appeared that my end results (whilst not as beautifully animated) had a little more clarity, so I'm glad I tried for myself. But this didn't seem like the end of the journey. Whilst this was an interesting experiment, and showed an lot of correlation between a sample of common fonts (as well as a couple of oddities – notably the lower case ‘g’ which clearly exists in two distinct common forms), what I really wanted was an average which somehow preserved the well-defined edges of existing fonts. So I started considering ways to produce a smoother, sharper average of letter forms.

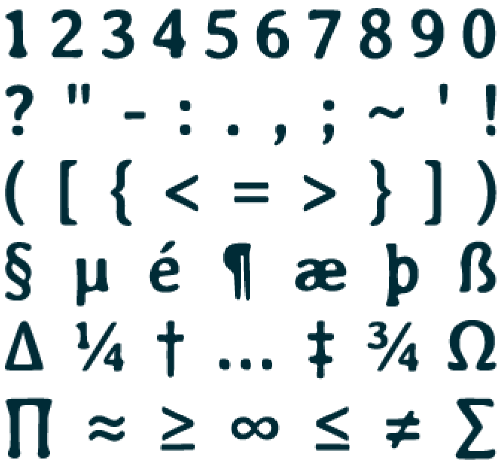
One idea which seemed obvious was to simply take the blurry results of the first experiment, and use a threshold to create monochrome images. A few experiments in this direction (I first tried with a lower-case ‘f’, which I later found was never likely to give good results due to the variance in height of the middle cross-stroke) convinced me that I needed to look into cleverer ways to achieve this. Surely there must be a simple way to average shapes, while keeping the result as a shape?



It turns out not to be straightforward. There are **many possible ways** to ‘morph’ between two shapes – and what might seem the most natural generally depends on our perception of ‘features’ in the shapes. Consider the average of a capital I with serifs, and one without: the natural thing to do would be something like, make the serifs half as big, and use a



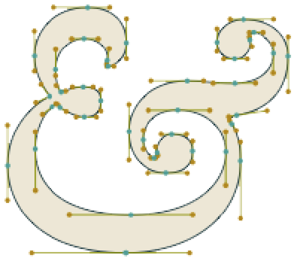




horizontal stem width about half-way between the two glyphs. That's two feature concepts being applied to the abstract forms[1]. To take a simpler example, what is the average of a square with the same square rotated 45°? There are a few possibilities …

I + | = ? ■ + ◆ = ?

So, this stumped me for a while. I decided I needed to get to know fonts better, so I built **a simple web app** to view the lines, curves and control points present in the fonts I had. On this basis, I started to consider the ways the features (vertices, curves, stems, serifs etc) might be matched up between fonts. However, this was a rabbit hole I might never get to the bottom of - particularly when considering some of the more unusual varieties of font. Perhaps there was a simpler idea that was evading me.



Then it occurred to me: since my aim was to average a large number of fonts, perhaps it would be best to use a very simple process, and hope the results averaged out well over a large number of fonts. So, how about splitting each letter perimeter into lots of (say, 500) equally-spaced points, and just average between the corresponding positions of each, on each letter? It would be necessary to match up the points so they were about the same location in each letter, and then the process would be fairly simple[2].

— · —

Having found a simple process to use, I was ready to start. And after about a month of part-time slaving away (sheer fun! Better than any computer game) – in the process of which I learned lots about bezier curves and font metrics – I had a result. I call it Avería – which is a Spanish word related to **the root of the word ‘average’**. It actually means mechanical breakdown or damage. This seemed curiously fitting, and I was assured by a Spanish friend-of-a-friend that “Avería is an incredibly beautiful word regardless of its meaning”. So that's nice.

— · —

Avería Bold ....

Avería Bold Italic ....

Avería Gruesa ....

Avería Regular ....

Avería Italic ....

Avería Light ....

Avería Light Italic ...

— · —

S A T O R
A R E P O
T E N E T
O P E R A
R O T A S

Along the way I naturally called on the counsel of the best designers I know – my brother **Nick Sayers**, **Lloyd Thomas**, **Tom Muller** and **Chris McGrail**, for advice. In the end, I decided to release the font using the **SIL Open Font License** – which means anyone can use it pretty much however they like – and to include within the family Regular, Bold and Light variants with Italics. Each is made from the corresponding subsets of the fonts on my machine. Also included is a "*Gruesa*" version made from all my fonts (725 in total).

Avería Family (ZIP, 369kB) [Updated 9 Nov 2011]
Avería at **The Open Font Library**

NEW by popular demand:
Avería Serif Family (ZIP, 323kB) **OFLB**
Avería Sans Family (ZIP, 320kB) **OFLB**

NEW Avería, Serif and Sans packaged as TTC TrueType collections (so you can install each family in one go, rather than one variant at a time). Thanks **Ludwig**:
Avería TTC Files (ZIP, 946kB)

NEW versions of Avería, based on **OFL** fonts from the **Google Web Fonts** directory - **now available through GWF as Avería Libre**:
Avería GWF Family (ZIP, 488kB)
Avería Serif GWF Family (ZIP, 432kB)
Avería Sans GWF Family (ZIP, 426kB)

Preview all

Feel free to **email me** if you have any questions – or use the **comments box below**.

N.B. I've had a number of emails from people asking if they can use Avería in various commercial / non-commercial projects. I'd love to hear if you do something with these fonts – but there's no need to ask permission. You are absolutely free to use them however you like.

—— · ——

Footnotes

1. In fact, there exist some interesting-looking typographic tools which do seem to offer the ability to interpolate between fonts – with an appreciation of the features which define the character of different typefaces. One is **Superpolator**; another is **Font Remix Tools**, for **FontLab**. Whilst I would love to play with these, they are rather out of the price range for an amateur like myself. Thankfully the open source **fontforge** was on hand and very useful for this project.

2. A problem I met along the way was how to deal with two particularly problematic cases – the lower case 'a' and 'g'. Both these exist in two different common forms – and the 'g' is particularly tricky as one variation has two holes, while the other has only one. After experimenting with various ways to compare the two and find a happy mean, I decided to simply choose the most common form, which turned out to be the one-hole g, the ‘round’ *a* for italics and the ‘hooked’ a for non-italic cases.

—— • ——

176 Comments **iotic** **Privacy Policy** 1 **Login**

♡ **Recommend** 188 Tweet Share **Sort by Oldest**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

jackassletters • 9 years ago

I love this font. I love the idea of it and am surprised you came up with something that is readable and pleasing. I'm not a font guy (as long as I have courier and helvetica I'm happy), but I like this.

5 ^ | ˅ • Reply • Share ›

Douglas Summers-Stay • 9 years ago

generative typography, huh? It sounds like Douglas Hofstadter. Great job!

6 ^ | ˅ • Reply • Share ›

nick ➦ Douglas Summers-Stay • 9 years ago

no, not at all actually.

4 ^ | ˅ • Reply • Share ›

Camilo ➦ Douglas Summers-Stay • 9 years ago

Maybe you could explain that. The only I've read from him is Gödel, Escher, Bach.

5 ^ | ˅ • Reply • Share ›

wtpayne ➦ Camilo • 9 years ago

Douglas Hofstadter has another book, "Fluid Concepts and Creative Analogies", in which (if I recall correctly) he documents his attempts to understand "fluid intelligence" by building analogy-making computational models. One of the models that he describes is a (simplified, restricted) font creating tool that takes a few example letters, and applies the same style to the rest of the alphabet. It has been a while since I read the book, but I do recall it being rather easier going than Godel, Escher, Bach.

6 ^ | ˅ • Reply • Share ›

Camilo ➦ wtpayne • 9 years ago

Gotta read that book (actually I'd read everything by him 5 times over if I had the time...), anything AI is such a good food for thought, and if you put fonts in the mix it's just double fun!

1 ^ | ˅ • Reply • Share ›

sgns • 9 years ago

Love it! Will try it out.

1 ^ | ˅ • Reply • Share ›

Luke Bradford • 9 years ago

Fantastic! Love the idea and the look of the font.

2 ⌃ | ⌄ • Reply • Share ›

Steve Jankowski • 9 years ago

Would you be able to post a list of all the source fonts you used to get this result?

7 ⌃ | ⌄ • Reply • Share ›

415raechill ➦ Steve Jankowski • 9 years ago

I would be totally stoked to see that!

3 ⌃ | ⌄ • Reply • Share ›

Al • 9 years ago

This is great. My friend Ganesh did a generative typography project a few years ago you might be interested in... it's a little flash app that you can play with in your browser:
http://ganeshnrao.com/#typo...

3 ⌃ | ⌄ • Reply • Share ›

Angel Ortega • 9 years ago

Beautiful! I only have problems with lowercase m. Thanks.

2 ⌃ | ⌄ • Reply • Share ›

iotic Mod ➦ Angel Ortega • 9 years ago

Yes, me too. I had to resist the urge to go in and fiddle with the glyphs manually – especially in that case. But I thought it would spoil the purity of the concept a little: I quite like that I haven't used the pen tool once. Of course, if a professional typographer fancied cleaning things up, that might be interesting ...

1 ⌃ | ⌄ • Reply • Share ›

Nils Werner • 9 years ago

Your font looks really beautiful when used in big letters like captions and headlines but unfortunately it's a bit distracting when used in body texts. Beautiful idea and name though!

3 ⌃ | ⌄ • Reply • Share ›

MrSpijker • 9 years ago

This is really nice!
Talk about a creative, average approach ^^
Job well done

1 ⌃ | ⌄ 1 • Reply • Share ›

Brandon Rhodes • 9 years ago

First, I am amazed that such a pleasantly scrappy-looking font is so readable — your blog post might be the easiest on the eyes of anything I have read so far this morning!

Second: how many fonts got averaged to produced Averia? I assume that since you are a font aficionado, you probably have quite the boatload of fonts installed on any given machine; but how many?

2 ⌃ | ⌄ • Reply • Share ›

iotic Mod ➦ Brandon Rhodes • 9 years ago

I did mention this at the bottom of the article, but you might have missed it – 725. That's for the *Gruesa* version, the others less as they are built from subsets (e.g. where the font metadata specifies italic and weight >= 500, for Bold Italic)

2 ⌃ | ⌄ • Reply • Share ›

Brandon Rhodes • 9 years ago

Brandon Rhodes • 9 years ago

(And, that gap in my last comment is actually encoded as an em-dash; will you be adding Unicode characters like that to the font at any point?)

1 ^ | v • Reply • Share ›

iotic Mod ➦ Brandon Rhodes • 9 years ago • edited

The em-dash is in the font. Maybe it's something up with DISQUS, or the way you entered it. Here's my attempt: "—".

The glyphs included are exactly the ones which were in more than half the fonts on my computer. You can see them all here (unlike this page, non-Cufón for non-Apple computers)

2 ^ | v • Reply • Share ›

Brandon Rhodes ➦ iotic • 9 years ago

When I hit “reload” I can see our em-dashes for a moment, while the page is rendered using one of my standard system fonts, then the em-dashes disappear as the page re-renders using your font. So Disqus seems to be delivering the character successfully; the problem must lie either in the font or in how Google Chrome is trying to interpret it.

1 ^ | v • Reply • Share ›

iotic Mod ➦ Brandon Rhodes • 9 years ago • edited

Ah, I think I know why. I take it you're not using MacOS or iOS. The page is currently using Cufon (don't dare to put the accent on the o!) to avoid nasty auto-hinting on Windows and Linux, and I made the cufon version with a smaller glyph set. You can see the @font-face version here (except on IE)

[edit] Cufón - now with all glyphs

2 ^ | v • Reply • Share ›

spookylukey • 9 years ago

Perhaps it would be best to avoid publishing which fonts you used, to dodge any possible litigation problems?

2 ^ | v • Reply • Share ›

Adrian • 9 years ago

Thanks for sharing! I'm surprised by how nice it looks and can imagine myself trying to use it in a project.

1 ^ | v 1 • Reply • Share ›

John Fullerton • 9 years ago

That is way cool! What a neat idea, and cool project to boot.

1 ^ | v • Reply • Share ›

Yuri Gordon • 9 years ago • edited

I described exactly the same process in my Book about the [Russian] Letters from A-a to Ya-ya.

I came to the conclusion that the basis of all the text fonts is a single structure, a kind of archetype. Every type designer tries, wherever possible, move away from this archetype, but he/she can leave only at a distance does not interfere with reading.

All text fonts - a product of the struggle between the creative will and the dictates of the reader.

My experiments can be found here: http://yurigordon.livejourn....

Unfortunately, in Russian :)

24 ^ | v • Reply • Share ›

24 ^ | v • Reply • Share ›

Cassandra ➦ Yuri Gordon • 8 years ago

Dear Yuri, your work is beautiful! I hope one day to get your book. It reminds me very much of the work of Robert Bringhurst.

1 ^ | v • Reply • Share ›

Neil Bowers • 9 years ago

Neat. Would be interesting to see an average serif font, an average sans-serif font, etc.

3 ^ | v • Reply • Share ›

Tee • 9 years ago

:)

1 ^ | v • Reply • Share ›

Kyle McDonald • 9 years ago

great results! it reminds me strongly of "flexure", which i best remember being used for "nintendo power" magazine in the late 90s.
http://www.google.com/searc...

but i'm a bit disappointed you didn't see all of the other excellent work people have been doing with generative/computational typography and generative font design in the last 10-15 years. there's some great work out there if you dig slightly little deeper.

i'm glad to see you went with the morphing technique rather than the averaging in the end. it's the difference between jason salavon http://salavon.com/work/Cla... and face of tomorrow http://www.faceoftomorrow.com/

1 ^ | v • Reply • Share ›

hawken • 9 years ago

I shall do this will all my Japanese fonts, see you in a decade!

3 ^ | v • Reply • Share ›

GHDesigns • 9 years ago

Nice!
It feels like all fonts together look like a 16th century derivative.

2 ^ | v • Reply • Share ›

Jim Fred Satko ➦ GHDesigns • 8 years ago

I was just going to say I like how it looks like (over-inked?) 18th century movable type a la Poor Richard's Almanack. :) In fact, I thought it had been designed as such, so it was very surprising to learn that your method of creation was entirely different. Nice! Jealous of your technical mastery.

^ | v • Reply • Share ›

Bombo • 9 years ago

Wow, big support. Awesome project.

1 ^ | v • Reply • Share ›

Khomuz • 9 years ago

Wow this is just epic. Well done you. Thanks for all your hard work. I will try to figure out a way to use this typeface somehow.

1 ^ | v • Reply • Share ›

mondaymorninglunatic • 9 years ago

Great Idea!

But there are some other great possibilities. You could create a font-space in

which every new font gives a new dimension for the coordinates.
3 typos would give 3 axis. each from 0 to 1.
One could have an n-dimensional font-space in which one can give an n-vector for a specific coordinate. in 3-font-space it could be like (1,2,4) for 1 part helvetica, 2 parts Times New Roman and 4 parts Comic Sans (HARHARHAR!!!).

Also things like koordinate-dependent typos would be great.
Start a page in the upper left with one font. the more you go to the lower right the more it becomes another typo.
Also seeing Bold and oblique as parameters more than options could be a great thing.

I am looking forward to this.
Really great work and great idea.

1 ^ | ˅ • Reply • Share ›

iotic Mod ➦ mondaymorninglunatic • 9 years ago

Have a look at Superpolator. Also multiple master fonts

1 ^ | ˅ • Reply • Share ›

Joel Dueck • 9 years ago

Might you see your way clear to cleaning up the kerning a bit? :-)

1 ^ | ˅ • Reply • Share ›

iotic Mod ➦ Joel Dueck • 9 years ago

The kerning is an average of the kerning pairs from the fonts that had kerning tables for the glyph set. It seems to me the process somewhat errs on the side of under-kerning. Are there any particular kerning pairs you'd like to bring to my attention as being off?

1 ^ | ˅ • Reply • Share ›

Joel Dueck ➦ iotic • 9 years ago

Ah, I suspected the averaging approach was applied to the kerning as well. I just think it would be a boon if the usual "obvious" culprits could be tightened up a bit -- combinations of V A T W Y L v w y, and lowercase letters a e o w v y s r u when following F K P T U V W.

I understand that kerning needs to be comprehensive in order to be consistent, in addition to being somewhat subjective, so I understand if this can of worms is too big to open right now. On the other hand, maybe a tweak in the bias of the kerning averaging would be a quicker fix, and more in keeping with the spirit of the experiment.

1 ^ | ˅ • Reply • Share ›

iotic Mod ➦ Joel Dueck • 9 years ago

I did some tests with increasing the kerning amount, and looked at changing the averaging methodology. In the first case, the kerning unfortunately seemed to be worse with an increased bias. In the second case, while I succeeded in cleaning up the averaging method a little it had negligible impact on the resulting kerning tables.

So I'm going to leave it as it is for now. Click here for a comparison of kerning using font metrics, Photoshop's 'optical' kerning algorithm, and no kerning - for Avería and some other fonts. Of the sample, I'd say only Avería is (arguably) better (in places) in the middle column, but I don't think the metrics suffer too badly in the comparison.

1 ⌃ | ⌄ • Reply • Share ›

Joel Dueck ➦ iotic • 9 years ago

Awesome, thanks for doing the comparison!

1 ⌃ | ⌄ • Reply • Share ›

nicolabricot • 9 years ago

Really great, i love it !
Thanks :)

1 ⌃ | ⌄ 1 • Reply • Share ›

Felix • 9 years ago

Have you tried this for some interesting subset of fonts? Would the result look drastically different, or is your font the analog of a brown color and always pops up if you mix sufficiently many?

3 ⌃ | ⌄ • Reply • Share ›

Marcel Bischoff • 9 years ago

I like the look. Reminds me a bit of the result which I got when I scanned a document and vectorized it with potrace

1 ⌃ | ⌄ • Reply • Share ›

gdorn • 9 years ago

I'll second the idea of generating Avería Serif and Avería Sans; one problem with Avería is is that the semi-serif aspect is particularly muddy. (I'm also not a typographer...)

1 ⌃ | ⌄ • Reply • Share ›

iotic Mod ➦ gdorn • 9 years ago • edited

Coming soon ...

[edit] Serif and Sans families now available - see bottom of article. Also main family slightly updated.

2 ⌃ | ⌄ • Reply • Share ›

Αντώνης Παππάς • 9 years ago

Did you do it by finding each font's r(t) curve function delineating each letter's surface for every letter and average them ? You can even get those brown points you have by doing a linear approximation of the letter's r(t). But you wouldn't be able to do have brown points in corners because there's no derivative there... :(Anyway would you please elaborate the algorithm used?

1 ⌃ | ⌄ • Reply • Share ›

iotic Mod ➦ Αντώνης Παππάς • 9 years ago

OK, here's what I did:

1. Turned all the fonts into SVG fonts using fontforge
2. Parsed all the SVG paths into separate perimeters, lines and curves
3. Centered each on its bounding box and normalized for units-per-em
3. Estimated the length of each perimeter using linear approximations for beziers
4. Went along each perimeter curve dividing into 200 points
5. Saved the data, ordering the perimeters by inner and outer, high to low and left to right

Then for averaging:

1. Take each glyph, use a 'model' font for 'correct' number of perimeters, and find all fonts having the same number of perimeters for that glyph
2. Make a cumulative average - each time aligning the points by rotating for a least squares best fit

3. Create SVG path from 200 lines
4. Compile all glyphs to make SVG font
5. Get averaged metrics for family and individual faces
6. Get average kerning tables
7. Simplify points to bezier paths with extrema
8. More simplification, cleanup and export to TTF in fontforge

6 ⌃ | ⌄ • Reply • Share ›

Avinash Arora • 9 years ago

Awesome story, and awesome end product!! I'm excited to play around with it :)

1 ⌃ | ⌄ • Reply • Share ›

Mike Burgess • 9 years ago

Very average

3 ⌃ | ⌄ • Reply • Share ›

Load more comments