Interview with Charles Bigelow

Yue Wang

Abstract

Interview of Charles Bigelow by Yue Wang, conducted in 2012.

Y: In this interview we are very lucky to have Charles Bigelow with us. Professor Bigelow is a type historian, educator, and designer. With his design partner, Kris Holmes, he created the Lucida family of fonts used in the human-computer interfaces of Apple Macintosh OS X, Microsoft Windows, Bell Labs Plan 9, the Java Developer Kit, and other systems, bringing historical and technical understanding of type to hundreds of millions of computer users. In 2012, Bigelow retired from the Melbert B. Cary Distinguished Professorship at Rochester Institute of Technology's School of Print Media. He is now the RIT Scholar in Residence at the Cary Collection, RIT's rare book Library.

C: Thank you for your visit.

1 Entering the digital type era — the birth of Lucida

Y: Let's get started. Can you briefly introduce the design goal of Lucida?

C: In the early 1980s, we saw that computers would become more widely used and that digital typography would be possible for more people. At that time, digital printers and computer screens had low resolutions. The goal of Lucida was to create a new, original family of fonts for medium and low-resolution digital printers and displays.

Y: Is this the reason why that's called Lucida?

C: Exactly. We wanted to give it a name that could suggest it was made of light and was clear despite the low resolutions. "Lucida" comes from the Latin word "lux" for light and clarity. It turned out that Lucida was the first original typeface designed for both digital printers and computer screens.

Y: Wow, really?

C: Yes. There had been previous digital typefaces designed for high-resolution typesetting machines in the late 1970s and early 1980s; a few were original types like Hermann Zapf's Edison, but none were new and original for laser printing and display screens (mainly CRTs in that era). Adobe developed their own font format called "PostScript Type 1" and digitized 35 typefaces for Apple's LaserWriter Plus printer. These fonts, including Helvetica, Times Roman, Palatino, etc., had originally been designed as metal type, and some like Zapf Chancery for phototypesetting. Designed before the digital era, those faces were not created for low-resolution digital rendering. When the first commercial font of Lucida was shown in 1984, it surprised Adobe. They knew of it; they had even digitized a test version, but they hadn't thought anyone would take the risk of making new designs for the new technology of laser printing. Instead, a Silicon Valley digital printer firm, Imagen, founded by Stanford researchers and graduates, some of whom had worked with or been students of Donald Knuth, brought out Lucida first. Imagen's type director, Mike Sheridan, wanted to produce a new design for the new technology and chose Lucida. Now, 30 years later, it appears he was right, but at the time, he took a risk. Adobe licensed Lucida fonts some years later and still distributes them.

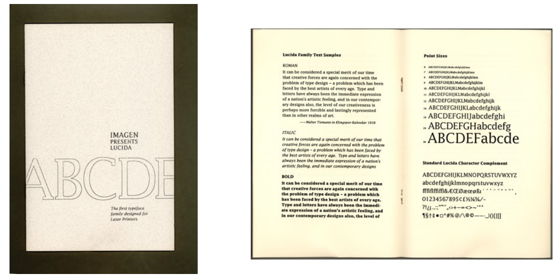


Figure 1: Lucida was the first new, original family of types designed for digital laser printers and screens. This is the first Lucida specimen, printed on a 300 dot per inch digital printer by the Imagen Corporation in California. Distributed at the ATypI conference in London, September 1984.

C: Here (fig. 1) is the first Lucida (seriffed) specimen, printed on a 300 dot per inch digital printer by the Imagen Corporation in California. It was distributed at the ATypI conference in London, September 1984.

Y: Cool. The specimen only included Lucida (seriffed).

C: Yes. The seriffed family was first shown in 1984, and the sans-serif family was released in 1985.

Y: What makes Lucida look great even in low resolutions?

C: We first did experiments, making bitmap letters by hand and comparing them to what we thought would be the outlines that could produce them. We found out several factors (see fig. 2). First, a big x-height packs more pixels into the most visually important portions of text, the x-height parts of letters. A big x-height is an advantage for texts read mostly on screens. That's one reason Apple has been using Lucida Grande as the standard user interface typeface on Mac OS X.

This interview was made possible with support from *Programmer Magazine* in China, `http://www.programmer.com.cn`.

Figure 2: Early studies for Lucida, comparing brush and pen written letters to bitmap redesigns for low-resolution printers and computer screen displays.

Y: That's why Lucida's x-height is bigger than most fonts, such as Times Roman or Baskerville, when composed at the same point size.

C: Exactly. There are still questions today about the importance of x-height for legibility in Latin alphabetic fonts. A vision scientist, Gordon Legge, and I recently wrote an article on the importance of type size for legibility, and we argued that x-height is the main factor that affects perception of type size [6]. The measure of x-height applies only to typography with upper and lower-cases: Latin, Greek, Cyrillic, and Armenian. For case-less writing systems, various other factors affect the impression of size.

Secondly, Kris Holmes and I observed that technical publications make frequent use of words in all capitals, such as acronyms, emphasized expressions, keywords, and the like. Therefore, we made the Lucida capital height a little shorter than the ascender height (e.g. the height of a lower-case 'h' or 'l'), to reduce the distracting look of words set in all capitals. This was not a new idea in typography; in 1495, the famous Venetian printer Aldus Manutius introduced a roman type with slightly shortened capitals cut by Francesco Griffo.

Third, the weight of Lucida is darker than traditional book typefaces. We noticed that on screens with black text on white backgrounds, the letters were slightly eroded, seeming too light, so we darkened the Lucida stem weights a little bit. The stem weight is 1/5.5 of the x-height, and a little bit less than 1/10 of the body size. Its overall gray tone is roughly 22% when the text is set solid (no extra line spacing).

Fourth, at low resolutions, a single pixel is often the only space between letters rendered at text sizes (8 point to 16 point). If letterspacing is tight, which was fashionable in advertising typography in the

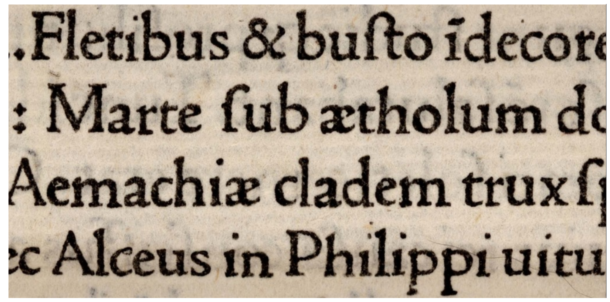


Figure 3: Scanned image from a book printed by Nicholas Jenson in Venice, 1478. These early typographic letters are rather dark and widely spaced.

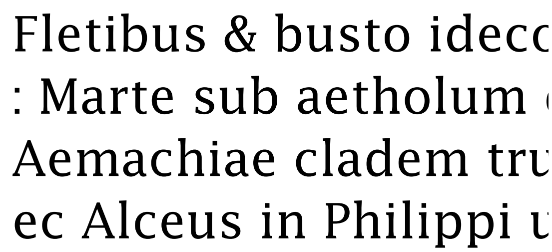


Figure 4: The same text from Jenson composed in the original Lucida font. Lucida is also somewhat dark and widely spaced, for early digital printing technology and display, but it is not a copy of Jenson.

1970s and 1980s, it can cause letters to touch. Some designers called this "sexy spacing", but it turns out to impair legibility. There are still debates about whether legibility is based on recognizing whole words or individual letters. Lucida is on the side of letter recognition. Computer screens were read from greater distances than print, which visually reduced letterspacing and caused crowding of the shapes, so we gave Lucida slightly loose spacing to counteract these tendencies.

Also, we created letter forms with large open counters — the internal open spaces like in 'a' and 'e' — to keep the interiors from collapsing and reducing legibility. Another small detail, which almost nobody notices, is that we lowered the joins of the arches in letters like 'n', 'm', 'h', 'r', and 'u', to give them more definition.

Y: So it won't clog up :).

C: Yes. For instance, we didn't want the top of an 'n' to clog up and look like a smeared 'o'. In fact, most of the ideas behind Lucida were not new. Some we borrowed from very early typography. Here's a scanned image from a book printed by Nicholas Jenson in Venice, 1478, when printing technology didn't have as high a quality as in later eras (fig. 3). These early typographic letters are rather dark and widely spaced, too. The forms are somewhat distorted by the technology of early printing. Rough paper, soft metal type that wears quickly, uneven pressure and ink squash, and so on. We borrowed some of Jenson's design ideas and believe we were the first to try them in the low-resolution digital era. Here (fig. 4) is the same text composed in the original Lucida font. It is not a copy of Jenson but shares similar goals — to make legible letters for a noisy medium. Jenson's type was around 15 point, but Lucida is more often used at smaller sizes — 10 to 14 point on screens — so we made its spacing even a little looser.

Y: Amazing!

C: Lucida, by showing some successful solutions to resolution-restricted text, also encouraged other designers to innovate. An interesting question is whether designers should try to compensate for limitations of new technology or design ideal shapes. Lucida is a design to compensate for limitations of resolution and imaging, but, in contrast, exuberant digital cursives like Zapfino or Apple Chancery are designs that take advantage of technological advances in character substitution.

Ten and twenty years after the first Lucida fonts, other designers created their own solutions to the problem of creating new faces for low resolutions. This is Lucida in 1985–87. In 1996, Microsoft released Verdana and Georgia by Matthew Carter, for the Windows operating system and Internet Explorer. Ten years after that, Microsoft released the ClearType font collection, including Calibri, Cambria, Candara, Consolas, Constantia, Corbel, and Cariadings (see fig. 5) by several designers, among them my former student, Gary Munch (Candara). These fonts take advantage of advanced screen display technology by Microsoft.

Y: They look similar to Lucida. Corbel and Lucida Sans are almost the same.

C: Well, but they are not copies of Lucida. These later designs show similar adaptations to the problem of design for screens: large x-heights, loose letter spacing, open counter-forms, and simplified letter shapes. In the alphabet samples at the bottom, the types are scaled to equal x-heights, to show similarities more clearly. Our emphasis on open counters and Renaissance forms for Lucida came from the calligraphic instruction Kris Holmes and I had with Lloyd Reynolds, calligrapher laureate of Oregon. Our idea of applying Renaissance forms to sans-serif came from Hans Meier's Syntax design, a sans-serif book typeface based on Renaissance humanist types and handwriting. It came out as metal type in 1968 and influenced not only us, but later generations of designers. The ideas of Syntax are now common in so-called "humanist" sans-serifs, but Syntax remains a splendid design, a great improvement on its successors.

A: Body sizes the same, x–heights vary

1984 Lucida, 1985 Lucida Sans, 1987 Lucida Bright

A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.

1996, Verdana, Georgia

A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.

2006 Calibri, Candara, Corbel, Cambria, Constantia

A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.
A quick brown fox jumps over the lazy dog.

B: x–heights equal, body sizes vary

Lucida Sans [12 point]

A quick brown fox jumps over the lazy dog.

Verdana [11.7 point]

A quick brown fox jumps over the lazy dog.

Corbel [13.7 point]

A quick brown fox jumps over the lazy dog.

Lucida Bright [12 point]

A quick brown fox jumps over the lazy dog.

Georgia [13.2 point]

A quick brown fox jumps over the lazy dog.

Cambria [13.6 point]

A quick brown fox jumps over the lazy dog.

Figure 5: Examples of original typefaces designed for digital printing and screens, showing convergent evolution. Lucida was the first original family for screens and laser printers. Later designs show similar adaptations to the problem of design for screens: large x-heights, loose spacing, open counter-forms, simplified letter shapes. In the alphabet samples at bottom, the types are scaled to equal x-heights to show relationships more clearly.

Y: I see. Why did Lucida Sans come out later than Lucida (serif).

C: We released the seriffed family first, as a kind of proof-of-concept, and then worked on the sans-serif. After the seriffed design came out, there was interest in the sans-serif version, which we released within the next year. Historically, because of the effort involved, large typeface families were not often released all at once. For a seriffed family, maybe only roman and italic would be released, and later one or more bold weights. If a typeface became popular, then a firm might release more variants (like more bold weights) or companion typefaces (like sans-serif, or fancy characters). In the mid-1930s, Jan van Krimpen in Holland was the first designer to create a family that included serif and sans-serif, and also chancery cursive variations, in his Romulus typeface family. The sans serif set was never commercially released, alas. Following Van Krimpen, we believed that a more harmonious pattern of text could be achieved if the different styles of type were designed together as an integrated set. This principle has held up well over the years. So, we created an extended family of serif, sans-serif, monospaced (typewriter), and various scripts (calligraphy, handwriting, casual), incrementally, over a period of several years.

Y: Because people use more variants in a single document? Knuth's Computer Modern is a complete font family too.

C: Yes. Donald Knuth, approaching typography with a mathematical intellect, also recognized the same principle that Van Krimpen first saw, that a typeface family could be implemented as a group of parametric variations of a basic form. Although Knuth says his goal was to imitate a metal typeface called Monotype Modern 8A, Computer Modern has many original ideas underlying its forms. In visual form, the basic seriffed version of Computer Modern did imitate Monotype Modern, but in conception and technical implementation, Computer Modern was original. It is noteworthy and commendable, too, that Knuth published all the Metafont code for his designs. For commercial reasons, most typefaces are marketed with intellectual property restrictions, but Knuth saw his typographic work as part of a greater goal, the publication of scientific literature and the dissemination of knowledge. He did the same with his TEX system for mathematical composition, publishing the source code for wide usage. A paragon of enlightened generosity.

An interesting aspect of Knuth's work on Computer Modern and the way he uses it in his TEX composition system, is that he established additional semantic categories for technical typography. Technical documents usually use different font variants to indicate different semantic meanings of the text and formulae. For example, in TEX, there are three slant variants — a slanted roman variant to indicate book names, a cursive italic to indicate emphasis, and a math italic for math equations. Prior to that, roman typefaces had either true cursive italic or a slanted roman (sometimes called italic) as their companion design, but not both. Times Roman, for example, has a cursive italic, but Helvetica has a slanted roman for italic. I am ignoring the slight visual adjustments that designers make to ostensibly slanted forms. In his work, Knuth began to use three slant variants, true italic, slanted roman, and math italic, and that led to us making the three italic variants for Lucida Bright math fonts: the normal text italic, which is semi-cursive; a slanted roman; and a cursive italic for math variables. To these, we added a chancery cursive in the 16th century Italian style, which we called Lucida Calligraphy. It can be used for math, but is more often used for display and ornamental typography.

This idea of a family of typographic variations is not new. It evolved over hundreds of years. In digital typography today, it is easier to produce typefaces than in the hand-cut metal era, so we can make bigger families within a few years instead of centuries.

One of the most fascinating trends in the history of typography is the development of new type forms and their incorporation into standard typography. Historically, roman capitals were used for inscriptions and formal handwriting during the Roman Empire (approximately 100 A.D.). Handwriting changed over the centuries and transformed the capitals into other styles that we now see as separate forms. Around 800 A.D., scribes working in the court of the emperor Charlemagne developed a "minuscule" handwriting ("small" handwriting) that had ascenders and descenders like today's lower-case. This Carolingian minuscule handwriting had no capitals. It was "mono-case" in today's terms. Around 1400, an Italian humanist scribe, Poggio Bracciolini, revived and combined the ancient roman capitals with the Carolingian minuscule to make a new kind of formal handwriting that he and other humanists thought was more legible than the gothic scripts then in wide use. (I should explain that these humanists were Italian Renaissance scholars and writers who shifted their studies from religion and theology, which had been medieval concerns, to philosophy, literature, classical languages (Greek and Latin), history, and other subjects we now call the "humanities".) Poggio made an amalgamation of what we now call upper- and lower-case in typography. A scholar friend of Poggio, named Niccolò Niccoli, developed a fast version of Poggio's handwriting. This was before printing; Niccoli copied many books, so he wanted a faster style of handwriting that was still legible. The Italians called Niccoli's style "running" hand(writing), "corsiva" in Italian. Today we use the term "cursive" in English to mean the same thing, a faster, freer script. The cursive tendency appears in other writing traditions as well. In Chinese writing, for example, there are both formal and cursive styles. You will know the Chinese names better than I do. On the formal side, there is Official style or clerical script (li shu) and a somewhat more cursive Regular style (kai shu). On the informal side, there is the semi-cursive style or running script (xing shu), and the very cursive style (cao shu).

To make a very rough comparison, Niccoli's cursive handwriting might be the equivalent of "xing shu". Some of Hermann Zapf's writing, like Zapfino, might be closer to "cao shu".

The first humanist roman types were cut around 1467, and the ancestor of most modern romans was cut in 1470 by Jenson in Venice. The first humanist cursive (italic) was not cut until 1501, and interestingly, it was cut in lower-case only. Its capitals were upright roman capitals. This shows that in those days, 500 years ago, capitals were not as tightly bound to lower-case as today. Also, cursiveness was not defined by slant alone, but by an ensemble of features, of which slant was only one. At first, italic type was an alternative to roman and whole books were composed in italic only. Italian calligraphers and type designers created many variations of italic, and later, in France, Robert Granjon cut many variations of cursive types. Around 1570, italic became a subordinate, complementary companion to roman instead of a stand-alone alternative to roman. Today, italic is an important component of a typeface family, but of secondary rank. In the 1700s, the French type designer Pierre-Simon Fournier expanded the concept of a typeface family to include variations with different widths and x-heights. In the 1800s, English typographers developed bold typefaces, which at first were separate from standard roman and italic, but by the early 20th century, especially in the designs of Morris Fuller Benton, some typeface families included bold weights as integrated members of the family. Thus, we see a pattern of incorporation of variations within a family. Sans-serif types were invented in the early 19th century but didn't become widely used for text until the 20th century. Looking at the pattern of type family evolution at the end of the 20th century, it seemed to us that incorporation of sans-serif into type families was a trend we should follow, and indeed, today in the 21st century, there are now several type families that include seriffed and sans-serif variations.

Adrian Frutiger is one of the most prominent figures in this movement to extend the visual range of type families. It is difficult to describe weight and style variations in words. We have to use cumbersome names like light, extra light, semibold, extra bold, ultra bold, light condensed, and so on, and the words are different in each language, so there are international communication confusions. For his large Univers family designed in the 1950s, Frutiger developed a two digit system to differentiate the weights, widths, and slants of the variations. The base of the system was 55, a normal weight roman, upright font. The first digit of the classification expressed the thickness of the weights, for example, 4 is light, 5 is regular, 6 is semi-bold, and 7 is bold. The second digit describes the style, for example, 6 is italic, 7 is condensed. So, 56 means normal weight, slanted, whereas 65 means roman, semi-bold, and so on.

Figure 6: From left: Lucida Bright, Lucida Casual, Lucida Handwriting; comparison of letter 'a'. All three designs have the same x-height.

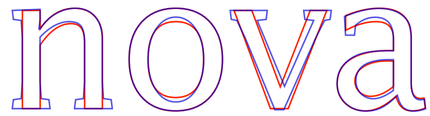


Figure 7: Comparison of original Lucida (seriffed) and Lucida Sans, showing close similarities of forms.

Y: It looks like a periodic table!

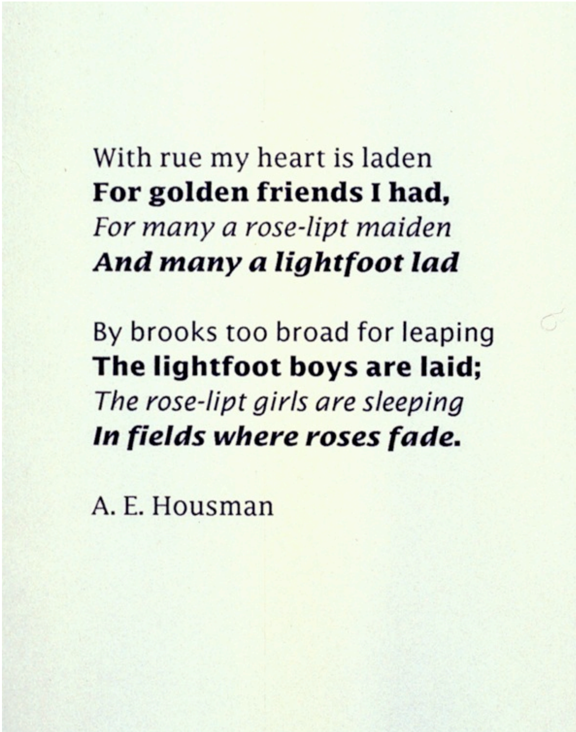
C: It sure does. So after Univers, designers were able to use many variants in a single document. Emil Ruder, a famous Swiss teacher of typography, demonstrated this in his book "Typography" [9]. Ruder's students continued this design approach. Thus nowadays we need large families of fonts for the most expressive kinds of modern typography.

Y: What are the common features among the big variations within the Lucida family?

C: In technical perspective, all the Lucida fonts have the same x-height, capital height, and similar series of stem weights, which helps give a harmonious look to a page that uses different font styles. Here (fig. 6) is a comparison of the letter 'a' in Lucida Bright, Lucida Casual, Lucida Handwriting. All three designs have the same x-height.

Y: I see.

C: But there are a lot of similarities among different font styles as well. For example, this (fig. 7) is the original Lucida Serif and Lucida Sans. The design is really harmonized. This (fig. 8) is an early specimen of the first four Lucida seriffed and sans-serif typefaces around 1986. But we didn't stop. We created an extended font family that included seriffed, sans-serif, and fixed-pitch (typewriter) designs. Around 2000, we had almost all the main variations of the Lucida typeface family of today. Here (fig. 9) is a list of them in normal form. As you can see (fig. 10), the typeface family is still highly unified and harmonized. The Lucida Bright family was developed



With rue my heart is laden
For golden friends I had,
For many a rose-lipt maiden
And many a lightfoot lad

By brooks too broad for leaping
The lightfoot boys are laid;
The rose-lipt girls are sleeping
In fields where roses fade.

A. E. Housman

Figure 8: An early specimen of the first four Lucida seriffed and sans-serif typefaces, circa 1986.

Lucida Bright *& Italic*
Lucida Sans *& Italic*
Lucida Casual *& Italic*
Lucida Fax *& Italic*
Lucida Calligraphy
Lucida Handwriting
Lucida Blackletter
Lucida Typewriter
Lucida Sans Typewriter

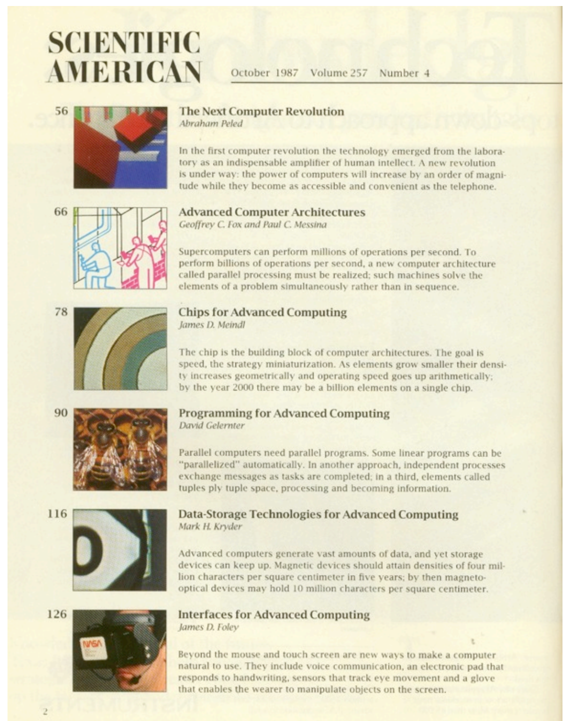
Figure 9: The main variations (in normal weight) of the Lucida typeface family, circa 2000.

for higher resolution systems, and was first used as the text face for Scientific American magazine in October 1987 (fig. 11).

Y: But for scientific journals there are also a lot of math equations.

C: Exactly. After Kris and I went to California, where I taught digital typography at Stanford in association with Donald Knuth, we wanted to make Lucida work well with TEX. Lucida's mathematical characters benefitted from the close association with Knuth. In

Figure 10: Study of differences in forms of different designs in the Lucida family.



SCIENTIFIC AMERICAN

October 1987 Volume 257 Number 4

56 **The Next Computer Revolution**
Abraham Peled

In the first computer revolution the technology emerged from the laboratory as an indispensable amplifier of human intellect. A new revolution is under way: the power of computers will increase by an order of magnitude while they become as accessible and convenient as the telephone.

66 **Advanced Computer Architectures**
Geoffrey C. Fox and Paul C. Messina

Supercomputers can perform millions of operations per second. To perform billions of operations per second, a new computer architecture called parallel processing must be realized; such machines solve the elements of a problem simultaneously rather than in sequence.

78 **Chips for Advanced Computing**
James D. Meindl

The chip is the building block of computer architectures. The goal is speed, the strategy miniaturization. As elements grow smaller their density increases geometrically and operating speed goes up arithmetically; by the year 2000 there may be a billion elements on a single chip.

90 **Programming for Advanced Computing**
David Gelernter

Parallel computers need parallel programs. Some linear programs can be "parallelized" automatically. In another approach, independent processes exchange messages as tasks are completed; in a third, elements called tuples ply tuple space, processing and becoming information.

116 **Data-Storage Technologies for Advanced Computing**
Mark H. Kryder

Advanced computers generate vast amounts of data, and yet storage devices can keep up. Magnetic devices should attain densities of four million characters per square centimeter in five years; by then magneto-optical devices may hold 10 million characters per square centimeter.

126 **Interfaces for Advanced Computing**
James D. Foley

Beyond the mouse and touch screen are new ways to make a computer natural to use. They include voice communication, an electronic pad that responds to handwriting, sensors that track eye movement and a glove that enables the wearer to manipulate objects on the screen.

2

Figure 11: Table of Contents of Scientific American, using Lucida Bright. Magazine redesigned by Bigelow & Holmes using the Lucida family.

fact, we continue to learn from Knuth's examples. This (fig. 12) is sample mathematical formulae with Lucida math fonts in 1992.

Y: Oh, this is an equation sample in Knuth's *The TEXbook*. It looks better in Lucida Math!

C: Thank you, but many people still prefer Computer Modern. Our Lucida math designs give users more choice, because the families look very different in text. With Berthold and Blenda Horn of Y&Y, we augmented the Lucida math character set with many more of the math operators and arrows in the Uni-

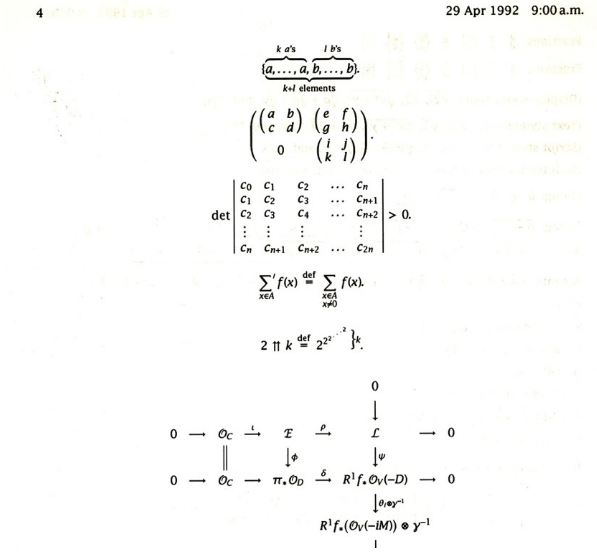


Figure 12: Sample mathematical formulae with original Lucida math fonts, 1992.

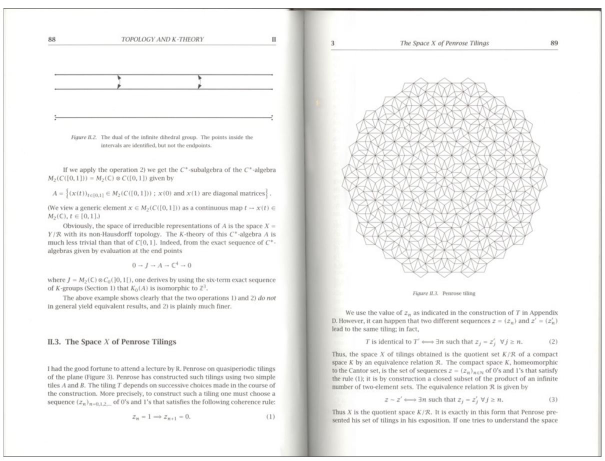


Figure 13: Lucida Bright and Lucida New Math in the design of the book *Non-commutative Geometry* by Alain Connes.

code character standard. Y&Y also developed many careful adjustments to ensure that the Lucida math fonts worked well with TEX. It was not an easy task. Here (fig. 13) is a book called *Non-commutative Geometry* by Alain Connes, which uses Lucida Bright and Lucida New Math. The book was designed by Peter Renz. Recently (2012) we upgraded the Lucida math fonts in cooperation with TUG, the TEX Users Group. We expanded the character set to include the latest Unicode math character blocks, including a new math script face by Kris, and the fonts were produced in OpenType format with the indispensable help of Khaled Hosny and others in TUG.

Y: So even without TEX, we can access these symbols using Unicode values?

C: Right. Because of Unicode encoding, computer fonts can finally contain a wide range of characters, letters, digits, glyphs, symbols, ideograms, logograms, and many others. You can include glyphs from various languages into the font. So we designed a lot of glyphs from various languages for Lucida Sans. This gave birth to Lucida Sans Unicode. We made Lucida Sans Unicode for Microsoft to show the possibility of what a Unicode font can do. Kris Holmes and I wrote a paper about this in 1993, which can be found on the web, "The Design of a Unicode Font" [2].

Y: Why make Lucida Sans Unicode? I remember Lucida Bright came out before Lucida Sans.

C: Lucida Sans was chosen to do this because of its popularity. For some typeface families that include both serif and sans-serif faces, one or the other is more popular. For Frutiger, the original sans-serif family is more popular than the Frutiger Serif, which is Frutiger's classic Meridien seriffed design re-worked and given additional weights and condensed italics by Frutiger and Akira Kobayashi and released in 2008. In contrast, with Palatino, the original seriffed design remains more popular than the very new and interesting Palatino Sans, by Zapf with Kobayashi, released in 2007. For text faces, it takes time for new designs to become widely accepted, so the balance of popularity between serif and sans could change in those families or ours. Every new, original type design is a risk because you don't know how well it will be accepted, and if you care only for acceptance, you don't give your design the fresh but risky insight that can make it popular. I like to quote the eminent physicist Niels Bohr: "Prediction is difficult, especially about the future."

Y: Also true for Erik Spiekermann's ITC Officina Sans and ITC Officina Serif which both came out in 1990.

C: Yes. A preference for sans-serif may be because sans-serif fonts are somewhat better for display on screens, probably because the sans-serif fonts are simpler in design, with fewer details, and therefore render slightly better at low to moderate resolutions. A vision study by Robert Morris, Kathy Aquilante, Dean Yager, and me [7] found nearly no difference between the legibility of seriffed and sans-serif typefaces when all the parameters (x-height, weight, spacing, etc.) are controlled — except that at small sizes on screens, sans-serif is slightly more legible. However, we did that study ten years ago, and although we controlled for resolution, today's new, higher resolution and higher contrast screen displays could perhaps alter our findings. I believe that serif types benefit from higher resolutions.

To cover more of the Unicode range for Lucida Sans, we designed characters for Unicode Extended

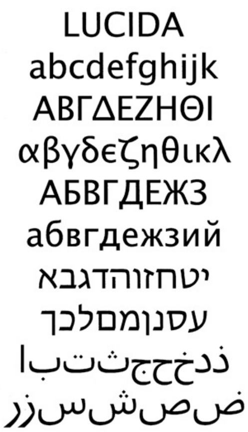


Figure 14: Lucida Sans (= Lucida Grande) non-Latin designs.

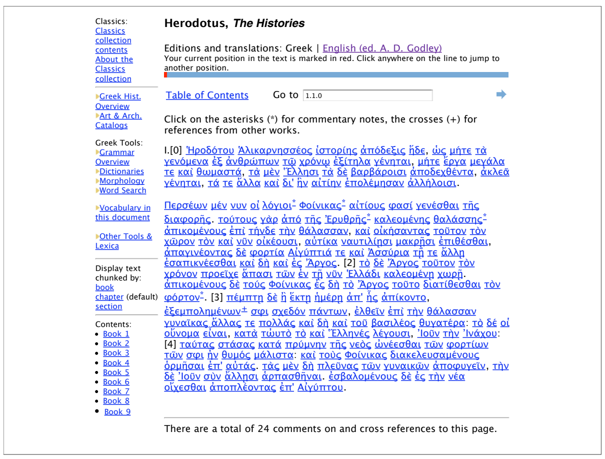


Figure 15: Herodotus text on-line (Perseus Digital Library): Greek text in Lucida Grande.

Latin, Greek, Cyrillic, Hebrew, Arabic, Thai, and other languages. After we did this for Lucida Sans roman, we designed extended Unicode sets for Lucida Sans demibold, and Lucida Sans Typewriter. This gave birth to the Lucida fonts used in the Java 2 developer kit in 1999 (see fig. 14). Starting in 2001, Apple's Mac OS X includes Lucida Grande, which is a further extension of Lucida Sans Unicode, as the main operating system font. For example, the on-line version of Herodotus (the first written "history" book in western civilization), released by Perseus Digital Library, uses Lucida Grande to display Greek text (fig. 15).

2 Research on digital rendering technologies

Y: It's amazing to see you follow so closely with the advancement of font technologies. Can you tell us more about how Lucida followed the development of font and rendering technologies? You said Lucida design was highly optimized for the screen. As digital fonts evolved from generation to generation, I guess Lucida changed too.

Figure 16: Pen written italic calligraphy by Kris Holmes compared to design of Lucida Bright Italic.

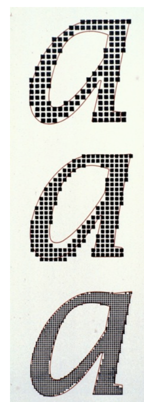


Figure 17: Comparison of bitmaps at different digital resolutions of original Lucida 'a'.

C: As I mentioned before, we did many experiments at the beginning. Here are the early studies of Lucida (see fig. 2). We realized that we should change some part of the shapes of the calligraphy to make it legible on a computer screen. For example, this (fig. 16) is the pen written italic calligraphy by Kris, but the glyph in the final digital font is different.

Y: At that time, most computer systems still used bitmap fonts.

C: Yes. On the Mac and Windows, screen fonts were originally stored in hand-tuned bitmap font files that specified individual pixel locations for a font at a particular size. We released bitmap Lucida fonts in various point sizes. Most times, for a given glyph outline, we marked every pixel inside the outline as black, and white for pixels outside the boundary. But this leaves ambivalences along the borders, so sometime manual fixes were needed to make characters more legible. Take the previous 'a' drawn by Kris as an example: here (fig. 17) is a comparison of bitmaps for different digital resolutions.

Y: I see. Then outline fonts were widely adopted, replacing the bitmaps.

C: Adobe was the pioneering digital publishing company at that time. They invented the PostScript language as their document format together with the PostScript outline font formats. Soon the PostScript language was widely adopted and PostScript's dominance seemed assured, and computer companies moved to adopt outline font technologies.

Y: So Adobe wanted to control PostScript to earn more money.

C: Yes, because they are a business. Adobe was in complete control of the PostScript technology at this point, and published an open PostScript language font format, called Type 3, but it didn't rasterize as well or as fast as Adobe's proprietary format, Type 1. A company had to license Adobe's PostScript to get Type 1 font technology, but major system software vendors like Apple and Microsoft didn't want a key font technology controlled by another company and didn't want to pay royalties for its use.

Y: So Apple developed their own scalable font technology.

C: Exactly. The code name was Royal, and later became called TrueType in 1991. The major technical differences between a PostScript font and a TrueType font, however, is that TrueType uses quadratic B-splines to represent the outlines, whereas cubic Bézier curves are used by PostScript. (See Robert Bringhurst, *The Elements of Typographic Style*, with a nice illustration on p. 183 in the third edition.)

Y: TrueType was a new technology. Why did it use a simpler representation (quadratic versus cubic)?

C: Some background. Several outline font formats were known then. Polygonal outlines — in which curves were approximated by a series of straight lines — were easiest to rasterize and been used for some successful digital typesetting machines, but needed too many points and were aesthetically inferior at larger sizes and higher resolutions, where the polygonal approximations of curves could be detected. Outlines composed of vectors and circular arcs needed fewer points and were fairly fast to process, but the radii of shallow arcs would be very long in comparison to the very short radii of small arcs. This problem was called numerical instability. Also, at high resolutions, there were noticeable discontinuities at tangents where a circular arc joined a straight line and curvature fell to zero.

When the outline description went beyond circular arcs and vectors, computer scientists tended to choose representations more on mathematical aesthetics than visual aesthetics. Peter Karow's Ikarus, the first commercially successful digital outline font development system, used cubic splines in Hermite form as a master format but for practical graphical output converted the Hermite cubics to circular arcs. Knuth preferred cubic splines and based Metafont on the mathematics of parametric cubics by Sergei Bernstein — also spelled Bernshtein — a Russian mathematician. Adobe chose cubic Bézier splines, developed for computer graphics by Pierre Bézier, also based on Bernstein's work.

Apple chose quadratic B-spline outlines in part because they already used them in MacDraw, a drawing program for Macintosh, so Apple had a proprietary outline technology in-house. Apple planned to use TrueType technology for the Macintosh user interface, so they wanted fast processing and believed that quadratic B-splines could be rendered faster than cubic splines.

A very interesting outline technology was developed by Vaughan Pratt, a computer scientist at Stanford, and used by Sun in a font format called F3. It was based on generalized conic curves [8]. Pratt's inspiration went all the way back to an ancient treatise on conics by a Greek mathematician, Apollonius of Perga.

I personally liked Pratt's approach best because it was a nice compromise between computational elegance, processing speed, and intuitive geometric understanding by designers. Sun did not push to establish their conic technology as a standard, so it was eventually overwhelmed by TrueType and Type 1. I tried to persuade some Sun executives to make it an open format and the standard in Solaris and Unix, but they apparently preferred to let it die than to let it out. Later, the Java language was saved from nearly the same fate.

With such a wealth of varied mathematical representations of fonts, it was difficult to tell which, if any, were artistically superior. Visually, the quadratic and cubic forms seemed more or less equally good at representing known type forms, so different firms chose font outline representations for engineering or commercial reasons or for non-visual mathematical aesthetics.

Y: So you created the TrueType version of Lucida using quadratic splines.

C: Yes. At this stage, Apple asked us to help them explore how to make things as simple as possible. We conducted a lot of experiments using Lucida. We went with Apple's font manager and chief font engineer to URW in Hamburg, Germany, where Peter Karow at URW had invented Ikarus in the 1970s. URW had over time developed a big library of digital outline fonts. To make TrueType successful, Apple needed a good supply of high-quality digital font data, and URW had the best and the most. They also had the technical ability to write accurate conversion programs from their format to TrueType. Most of the early TrueType fonts were produced from Ikarus format data, including the Lucida fonts, because we used the Ikarus system to digitize our designs. One of the most important experiments was, how to use as few points as possible to represent a font outline. If we had fewer points, font file sizes were smaller and, importantly, computers could render fonts faster.

Y: This is also true today. Today most of the graphics and animations are offloaded and processed using special hardware in the computer. So ironically text rendering is even slower than graphics display.

C: That's interesting. Text has some advantages over general graphics, at least for alphabetic fonts, because there are relatively few characters, so once they are rasterized for a given size and resolution, the rasters can be cached and fetched very quickly, so the pages are essentially tilings of a small number of stored and repeated graphical elements. For Chinese fonts, however, the characters are more complex and many more of them are needed, so processing was still a problem until recently. In those early days, in addition to limited processing capability, we also had other problems. Computers had limited memory, and most people were still using floppy disk. Though quadratic splines use fewer parameters than cubic splines, we needed more points to represent the shape well. To save memory, it was important to use as few points as possible, but you cannot use too few of them or the glyph outlines will be distorted. Kris and I did a lot of experiments to show Apple how many points to use when creating a font outline.

There is a particularly interesting problem with TrueType splines when the number of points representing a curved quadrant is reduced below some threshold. The shape of the curve bulges out at the corner, and a quadrant of a circle or ellipse becomes hyper-elliptical, to use a term by Piet Hein. This is a subset of a general question about mathematical representations of shapes that were created by motions of the human hand. When Donald Knuth was working on Metafont at Stanford, he would meet with interested students and colleagues at lunch to discuss a wide range of questions and problems that came out of his research. He called it the "Metafont for lunch bunch". We discussed how the mathematics of the equations affected the forms of the curves in typefaces, and we wondered what kinds of curves were sufficient for representing the aesthetics of traditional typefaces. I am not a mathematician, but I found those discussions fascinating because Knuth was leading all of us into a barely explored realm where mathematics and aesthetics met.

Today, a quarter century later, most computer-aided drawing programs and type design programs uses Bézier cubics, and sometimes I see a tendency for recent typefaces designed directly on the computer to seem similar in the modeling of forms. I believe that this is the result of interaction between vision, user interfaces, and mathematics. Bézier splines can behave in surprising and anti-intuitive ways, at least for artists accustomed to drawing and writing on paper, and they don't necessarily resemble the motions of the human hand. When designing on screens and using a mouse instead of a pen or brush, type artists tend to be conservative, using a small number of points on the curves and adjusting the off-curve control points carefully to make smooth shapes that are easier to understand and control. The curves are usually pleasant, but they are more limited than the shapes that result from the living hand moving a traditional tool through a complex path. The motions of the tip of a Chinese calligraphy brush are especially complex and subtle, for example.

Y: So, Apple asked you to help them solve very practical problems.

C: Right. Apple, Adobe, Microsoft, and a firm called Imagen, founded by two Stanford computer scientists, asked us for advice and consultation on various font technology and aesthetic issues. In the late 1980s, Apple invited us to do some new experiments. As I said, naive algorithms for rasterization cause various aesthetic problems on computer screens and low resolution printers, like irregular stem thicknesses and spacing, irregular letter heights, loss of serifs, broken hairlines, and so on. So when Adobe developed the PostScript font format, and later when Apple developed the TrueType font format in 1989–1991, font hinting was introduced to solve those problem.

I should explain that "font hinting" is the use of computer program instructions to adjust the display of an outline font so that it lines up with a rasterized grid. At low screen resolutions, hinting is critical for producing a clear, legible text. Hinting can be generic for all sizes, but TrueType hinting also has the capability of adjusting hints for specific resolutions. This localized or hand-tuned hinting has to be done by people, who can test and view different approaches. It has become a special skill practiced by a small number of experts. A typical kind of hinting problem is to make all the vertical stems of a font have the same pixel thickness, so the text looks regular in tone and rhythm. At a given size, the mathematical thickness of a stem might be, let's say, 2.5 pixels. So, depending on how a letter falls on the raster grid, a stem might be 2 pixels or it might be 3 pixels thick. This makes for a splotchy, irregular image. With hinting, all the stems can be forced to be 2 pixels, or 3 pixels, to enforce regularity. The actual outlines are being distorted, but the results look better to readers.

Y: So Lucida has hinting instructions inside the font file?

C: Right! In fact Lucida Sans roman was the first fully hinted TrueType font in history. Apple developed the format but didn't completely hint a font. At that time there weren't mature tools for hinting, and Apple didn't have type designers on staff, so they asked us to test the format by hinting a font, using low-level tools developed for programmers to write hinting code. Kris Holmes hinted a whole font that way. It was a lot like writing macro-instruction code. Kris showed that TrueType hinting worked in a practical design context, but the experience also made us realize that hinting was a separate kind of task from designing. We decided to stick to designing forms, not hinting them.

Y: Amazing! But I heard that hinting is not used in Apple's system any more.

C: Yes. Increasing resolution screens and new font rendering technologies, often called "anti-aliasing" or "smoothing", eventually made hinting unnecessary on later generations of screens and printers. That took more than a decade of progress, because display and printing technologies improve much more slowly than the rate of Moore's law. By the late-1990s, grayscale and color display screens gained enough market dominance that rendering algorithms could take advantage of the range of gray tones available on screens.

Y: Is this related to using anti-aliasing techniques from computer graphics?

C: Yes, the term and technique come from computer graphics. For a given glyph outline, the edge of a contour usually does not fall exactly on a pixel boundary. An edge pixel might be partly inside the contour (black) and partly outside the contour (white). Anti-aliasing adjusts the gray tonal value of that edge pixel in proportion to how much of the pixel is inside the contour or black area. The resulting edge looks smoother because the intermediate gray tone is not as noticeable as an all-black pixel. This method works better at higher resolutions. Below 100 pixels per inch, viewed at a normal reading distance, the result looks fuzzy or blurry. Above 200 pixels per inch, the result usually looks smooth without objectionable blur. In between, the reader's impression of sharpness or fuzziness depends on the display technology, such as LCD or e-ink, the contrast, the reading distance, and other factors. On very high resolution screens, like the Retina screens of iPhones or iPads, the edges look smooth and sharp because the human eye usually can't perceive lower contrast individual pixels at those resolutions. Vision scientists have measured the sensitivity of the human visual system to contrast and detail and found that as detail gets finer and contrast gets lower, it is harder and harder to see fine features like tiny pixels. Conversely, for fine details to be seen, they have to be high-contrast.

Y: What about ClearType?

C: As color LCD screens with resolutions above 120 pixels per inch gained in the market, subpixel anti-aliasing became feasible. Most computer color displays use pixels made up of three subpixels: red, green, blue stripes. Usually, each subpixel has 8 bits of tone value, equalling 256 possible gray levels. A white pixel has all three subpixels turned on, while a black pixel has all three subpixels turned off, and other RGB tone values produce millions of colors in-between. Because the subpixels are adjacent, a clever hack is to represent different spatial positions and line thicknesses by choosing different colors for the whole pixel that will turn on or off selected subpixels. Microsoft developed this concept into their "ClearType" technology in Windows. It effectively triples pixel resolution in one direction, because it uses sub-pixels, which are 1/3 of a full RGB pixel.

Subpixel anti-aliasing works better at resolutions above 150 pixels per inch, where color fringe effects become nearly imperceptible. At resolutions above 300 pixels per inch, the color effects are imperceptible, and resolution seems very sharp. It is also important to note that subpixel anti-aliasing works in only one direction, either horizontally or vertically, depending on the orientation of the RGB subpixels. For better resolution of letter stems and bowls in Latin alphabetic type, the RGB sequence should be oriented horizontally. For Chinese, which has more horizontal strokes, better resolution is obtained when the RGB sequence is oriented vertically. However, devices like iPad and iPhone can display in both orientations, so it isn't possible to optimize the character forms for one orientation.

Y: What's Apple's counterpart of ClearType?

C: Apple uses similar techniques in OS X and iOS, but without a trademark name. I assume that because of the cross-licensing of TrueType font technology between Apple and Microsoft, Apple can use subpixel rendering algorithms like ClearType without infringing Microsoft's methods or patents, but that's just my guess. However, ClearType is Microsoft's trademark, so that is presumably why Apple doesn't use that name. Apple's Retina displays use both high resolution and anti-aliasing.

Y: So you also need to think about how subpixel rendering affects the display of Lucida.

C: We can think about it, but it is hard for designers to do much about it. Subpixel rasterizing of larger type sizes on high resolution screens, which now have a major share of the market, needs no special efforts by designers because the edge artifacts from rasterization, including jagged staircase patterns, fuzzy contours, and color fringing, are small in comparison to the size of the letters and do not appreciably degrade the quality of the text image. Below 14 point, and at lower resolutions, type size is small relative to the sizes of pixels, so the rasterization artifacts are big in comparison to letter details like serifs, hairlines, and stems. The artifacts are noise that obscures the signal of character shape. In extended texts, there may be thousands of characters on a screen, so en masse, the artifacts can make text visually "uncongenial". Readers may not like the look of the text, though they may be able to read it nevertheless. Vision scientists have shown that low-resolution or fuzzy text can often be read as quickly and accurately as sharply rendered high-resolution type. The care that designers put into the shapes of characters, and the ingenuity that engineers put into rendering technology, contribute more to aesthetics than to legibility. Type is both aesthetic and informative. Well-formed and well-rendered text contributes to the pleasure of reading a text.

Recognizing the importance of designing for subpixel anti-aliasing of text types at text sizes, Microsoft commissioned several new, original font families to work especially with ClearType technology. Several were for Western scripts — Latin, Greek, and Cyrillic — and one was for Japanese Kanji, Kana, and Romaji. We tested them in my course on news typography at RIT a few years, and they all looked good. I was happy to see such strong support of artistic creativity for a new technology, from a major technology company. I should say that one of the ClearType font designers was a former student of mine, and others were friends, and that Microsoft also licenses Lucida fonts, though not as part of the ClearType set.

I think the next big challenge for designers of text type will not be pure legibility, although that is the worthy goal of most text face designers and is achievable with existing designs in current rendering technology on high-res screens. Instead, I expect to see more emphasis and experimentation with expressiveness in design, coupled with congeniality for the reader. In the past five centuries of development, Latin alphabetic typefaces have become highly refined in their forms, weights, patterns, and variations, and many have proved to be legible over centuries. More than half of the new novels published in the US in the past decade were composed with "Old Style" type designs based on typefaces first cut more than 250 years ago. Some of the designs, like various faces based on those by Garamond and his contemporaries, were first cut more than 450 years ago. So, at least for print book readers, the great old seriffed fonts of the past are still the great new fonts of today, in digital format.

Digital design tools and rendering enable greater precision and regularity in type forms, but the risk is that the designs look boring — too regulated, too repetitive, too rigid, too homogenized. Randomly adding irregularity doesn't improve the appearance — the designs then look boring but awkward. Some graphic and interface designers want neutrality in typography, but I don't believe that any type design is truly neutral. Every typeface carries some degree of expressiveness, even those intended to be plain, simple, and neutral. For example, a user-interface in Helvetica expresses a different feeling than one in Lucida, but the two designs are similar in weight and x-height. Helvetica is more modernist, Lucida more humanist. Helvetica more carved, Lucida, more handwritten. Helvetica more tightly spaced, Lucida more open. A Swiss poet made a memorable comparison of the feeling of Helvetica compared to Syntax Antiqua, a very readable sans-serif typeface by Hans E. Meier, which is even more closely based on humanist handwriting and early Renaissance typography than Lucida. The poet said, when he reads a page in Syntax, it is like walking through a field of flowers, but when reading a page in Helvetica, it is like walking through a field of stones.

So, a problem for future designers will be: how much expressiveness to put into a type. What expression does the design convey to the reader? For the reader, highly expressive typefaces are lively but can look too complex for long texts. Free scripts can look graceful but may seem too undisciplined for modern readers accustomed to rigidly regular fonts.

When technology changes, there are opportunities for new designs. We can find many historical examples. More than 50 years ago, typography shifted from metal to photographic technology. Hermann Zapf's Optima, first created for metal typography, became wildly popular in photo typography because it gave greater expressiveness to the sans-serif genre.

Optima's subtly flaring terminals and classical letter structures brought a hint of Renaissance proportions and humanist handwriting into a modern idiom, through a new technology that crisply reproduced designs photographically and lithographically, without the usual wear and ink squash of metal type. Yet, the subtle qualities that made Optima so successful in photo technology were difficult to render in early digital typography because of low resolutions on screens and printers, so Optima lost popularity in laser printing. Instead, Zapf's Palatino gained popularity in desktop laser printing because it conveyed some of the handwritten vigor of Renaissance typography and calligraphy even in low resolution of 300 dots per inch. Today, as digital resolutions increase, Optima is regaining popularity for a new generation of graphic designers. We may see new type designs for screens that enjoy similar popularity in the new media of e-books, smart phones, and pad computers.

I believe that expressiveness is also an interesting challenge for East Asian scripts. Chinese type styles derived from woodblock printing, like the Song/Sung styles, were adapted to metal typography and are now widely used in many variations in a large range of sizes in digital typography The same is true of the related Mincho styles in Japanese typography. The rectilinear structure of this type genre, which may have made it easier to cut in wood, makes it seem stiff and rigid but functional. It may be that Song style was easier to cut and cast in small sizes of type, which would have made the style more economical because small type sizes use less paper, and are thus more widely used.

Typefaces based on brush-written Chinese scripts have more handwritten grace but historically were more difficult to adapt to metal typography, and probably that is why they are less popular than Song or Mincho styles. Digital typography relaxes the technical limitations on producing and printing fonts, and makes it easier to "draw" digital characters, so we are beginning to see more expressive styles in Chinese and Japanese typography, but mostly for "display" in advertising, headlines, and other contexts, at relatively large sizes. Many of the recent fonts are not in traditional calligraphic styles, but are fanciful designs, like clouds, fat fish, childish writing, blurred writing, and so on. Perhaps some of these were already known in hand-painted signs and banners, and now can be made into type. If "folk" styles are getting made into type, that is fascinating. In American music, folk styles went mainstream because of the recording industry and we got jazz, rock 'n' roll, and country and western musical genres, which have since gained worldwide popularity. However, America has not produced a "breakthrough" folk typography, probably because lettering art, calligraphy, and typography have not been folk cultures, but the practice of literate elites. The ancient literate traditions of China, Japan, and Korea may, however, include styles of writing that could become newly popular in digital form. And, of course, young designers do not always want to follow old traditions, and instead invent new styles.

I think this is an exciting challenge for designers in China, Japan, and Korea — to capture the expressiveness of classical styles and adapt them to newer technology, without seeming quaint, old-fashioned, or reactionary, and to find interesting historical styles worthy of revival, but also to invent new styles. These trends are already happening in display types, used in large sizes, but the big challenge is, how to produce those kinds of expressiveness in text types that can be read at small sizes.

In English language book publishing, sans-serif fonts are very rare in literature of any kind, whether important literature or popular genres like crime, romance, and science fiction. Fiction is generally seriffed. Books about graphic design, photography, and modern art, however, use sans-serif types fairly often, so the choice of type style depends on the content and on the reader. I wonder if similar distinctions occur in East Asian publishing.

The recent popularity of Japanese "cell phone novels", which are usually about the lives of young people and often written by young women, are said to use more hiragana characters than traditional Japanese literature. I wonder if this increases interest in expressive hiragana fonts, when cell phone novels are published in print. There are already many expressive kana designs, which can be combined with appropriate weights and forms of Kanji to achieve subtly different text effects. When there is a shift in literary taste, there can also be a shift in typographic taste. Another interesting mixed writing system is the Korean, which uses Hanja characters based on Chinese, along with the unique Hangul alphabet. Compared to the Latin alphabets, Hangul more accurately represents the significant sounds of speech. So, I wonder whether literary expression that favors Hangul motivates trends in the graphical design or usage of Hangul fonts. Do font styles reflect literature? Are Korean pop novels and cell-phone novels using more Hangul than Hanja characters? The Korean Hangul writing system was sans-serif in early examples, but late styles became similar to brush-written characters.

Y: What about different weights in Lucida?

Figure 18: Spectrum of possible weights for Lucida Sans. Top group = "light" weights; second = "normal" weights; third = semi-bold weights; bottom = "bold" weights.

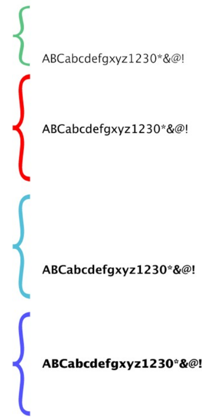


Figure 19: Current weights of Lucida Sans. Assuming stem weight of "normal" = 1.0, then "light" weight = 0.75 x normal; "demibold" = 1.5 x normal; "bold" = 2.0 x normal.

C: Here (fig. 18) is a series of experimental weights for Lucida Sans. The top group is for light weights, the second group for normal weights, the third for semi-bold weights, and the bottom group for bold weights. In the first generation of Lucida fonts, the low screen and printer resolutions could not support such fine gradations of weights, so we made only a few weights: normal, demibold, and bold. Now, higher-resolution display technologies and anti-aliasing techniques can render finer weight gradations, so we have designed additional weights of Lucida Sans, to be released next year. By studying the weights of popular text typefaces today, and also going back hundreds of years, we concluded that there is no single ideal weight, but a range of preferred weights, depending on printing quality, reading conditions, and, in digital displays, screen technologies.

At RIT, I did a study of "just noticeable differences" in the weight of a sans-serif face. For a given weight, how much darker must a slightly bolder weight be for a reader to notice that it is darker? The results appear to follow the Weber-Fechner law in psychophysics, which says that perception of difference is proportional to stimulus. I found that for a "normal" font of a certain weight, a just-noticeably darker font needs to be approximately 2.5% bolder than the normal weight. The same is true for a bold weight: the next perceptibly darker weight must be 2.5% darker than the bold, so perception of weight difference follows a geometric progression.

Y: The weight spectrum reminds me of the Frutiger numbering system!

C: Yes, Frutiger was a pioneer in the numbering of typeface weight systems with his Univers family and later with his Frutiger family and others. He saw that typographic weight nomenclature was a confusing mess. Different designers, type foundries, and font vendors used different and incommensurate names. Frutiger rationalized weights within Univers and designated them with two-digit numbers. I always liked that. Recently, a three-digit numbering system has been developed for Univers, to incorporate additional weights and widths. It is useful but doesn't exactly match the original two-digit system, which makes it confusing for me because I remember the older, simpler system. Around 20 years ago, Peter Karow, developer of the Ikarus software for type digitization, made an interesting study of the statistics of typeface weights, using a large digital font database. He made a reasonable proposal for rationalizing typeface weights in an 11-step system but it was not adopted. Today, W3C recommends a set of font-weight names and associated numerical values in a 9-step system, but it is, to my mind, inconsistent with existing progressions, arbitrary, and too limited, so I don't see it as an effective solution. I'm afraid, it is a muddle that won't be cleared up soon, if ever.

Y: What's the current weight of Lucida then?

C: For Lucida Sans normal, the stem thickness is 18% of the x-height. Lucida Sans demibold is 1.5 times the normal stem, and the bold stem is 2.0 times the normal. (See fig. 19.) This approximates a progression based on the square root of 2. However, weight measured by ratio of stem to x-height, which designers like, is not the same as weight measured by percentage of black pixels in total text area, which an engineer might prefer. Using pixel area weight measure, Lucida Sans normal is roughly a 22% gray tone. Lucida Sans demibold is approximately 29% gray, and Lucida Sans bold is 36% gray, which is 1.6 times the normal weight. Thus, the gray tone progression does not increase as much as the stem-weight to x-height ratio, because of the way weight is distributed in a Latin typeface — more of the weight is in the x-height region, less in the ascender and descender region.

Text typefaces appear to cluster into weight groupings. The normal weights of seriffed roman text faces tend to have light gray tones, ranging from around 14% to 18% gray. Seriffed types designed for screen display tend to be somewhat darker, around 18% to 22% gray tone. Sans-serif fonts for print and screen also tend to be darker, ranging from 19% to 23%. Of course, there are lighter and darker weights in many typeface families; I'm talking about what are called "normal" or "regular" roman text weights.

As a side note, Chinese fonts also cluster into tonal groups, but to measure the average gray tones is challenging, because the number of strokes in a character and therefore its density varies much more than in Latin typefaces, and the frequency distribution of characters can vary according to content and usage. In my very rough estimates, Song style faces have average gray tones that cluster like traditional seriffed Western fonts, but slightly darker, ranging from 15% to 20%. Sans-serif Chinese fonts tend to be darker yet, ranging from roughly 22% to 35% gray tone. However, I guess that weights darker than 30% are not often used in running texts. Kanji fonts cluster into similar tonal groups. I hope that type scholars in Asia will explore some of these patterns of usage.

Back to Lucida — to make Lucida Grande work well in Apple's OS X font menu, Apple preferred the designation "bold" to "demibold", so Lucida Grande Bold in OS X is the same weight as Lucida Sans Demibold in Windows. I regret the confusion — another difference between operating systems and platforms. Weight measurements, names and numerical values remain an unsolved problem of lack of standardization, in part because of the technical needs of various systems, and in part because designers simply make weights the way they think looks best.

3 State of the art — smart fonts

Y: Interesting. What other new technologies are you involved in when designing typefaces?

C: Apart from computer graphics techniques and higher resolutions, an important font technology is the glyph substitution technique used in OpenType.

Glyph substitution makes math fonts less cumbersome because different forms and sizes of glyphs can be substituted according to context. In Arabic typography, smart fonts are aesthetically functional. They enable easy use of context-sensitive

Figure 20: Demonstrations of the joining structure of Lucida Handwriting compared to Kolibri, a script design by Kris Holmes for URW, developers of the Ikarus font software used by Bigelow & Holmes.

shape variations that are aesthetically necessary in Arabic scripts. This encourages artistic expression and experimentation, both in capturing traditional styles and in imagining new styes. In terms of glyph variations, Latin alphabet fonts were simplified during the first hundred years of typography, with most ligatures, abbreviations, and alternate forms eliminated for economic reasons. So, smart fonts are not crucial for Latin alphabet typography, but do have artistic and ornamental value. Hermann Zapf's Zapfino, a graceful yet free script with glyph substitution, has become very popular. Some of Kris Holmes' scripts like Apple Chancery, which has many glyph variants, and Kolibri, which has intricate joining, also show the aesthetic possibilities of smart fonts. Jim Wasco's Elegy script also shows elegant use of OpenType.

Before OpenType, Apple invented a similar technology called TrueType GX, later called AAT. The software that renders text parses the strings for certain combinations and contexts of letters, and, when they are found, the software substitutes alternates from the font if the substitutions have been programmed into the font. A common example in English and European languages is the f-ligatures. To keep the dot of the letter 'i' or the top of the letter 'l' from bumping into the upper arm and terminal of the 'f', typefounders used to cast special combinations of 'fi', 'fl', 'ffi', and 'ffl', and more rarely, 'fj', for words like "fjord". A few like 'fi' and 'fl' are common in most fonts today. When we were designing Lucida, glyph substitution wasn't available so we designed the 'f' with a short top arm that didn't collide with the 'i' or 'l'. In Lucida Grande, several f-ligatures are available, like 'fi', 'fl', 'ff', 'ffi', and 'ffl'.

Kris continued to experiment with more complex character sets. We designed Lucida Casual with three alternative styles, though two of them have not been released because we were experimenting to see if glyph substitution made sense for them. However, glyph substitution is often not necessary, even for a lively script. You can see that in Lucida Handwriting. Kris crafted it so all the end strokes were placed in a single horizontal line. It looks like free handwriting, but has a simple joining method. As another example of alternative forms, Kris worked with Peter Karow at URW to design the Kolibri script, which has a more complex joining pattern than Lucida Handwriting, so that every character can join elegantly, but that requires many alternate forms. URW++ has now produced it in OpenType (fig. 20).

Y: Amazing experiment. So now we have four different 'e's. This script is really elegant.

C: Yes. Kris has a special liking for lively script faces. (I think I am permitted to boast on her behalf!) She studied dance and choreography for years as well as studying calligraphy, so her type designs learned many things from choreography, especially a feeling for motion and rhythm but also a sense of order within complexity. When Apple created a "smart" font technology based on their TrueType, it was first called QuickDraw GX in the mid-1990s but later evolved into Apple Advanced Typography, or AAT. In AAT, there can be several degrees of ligature control, old style figures, small caps and drop caps, swash variants, and alternative glyphs.

Y: This sounds very similar to Microsoft and Adobe's OpenType.

C: Exactly. When Microsoft wanted to use Apple Advanced Typography, Apple refused to license it, so Microsoft and Adobe worked together to create OpenType, which is technically somewhat different, but provides much of the same functionality. But back to QuickDraw GX and AAT — when Apple was developing the new font technology, they showed us a page like this (fig. 21). It's chancery cursive writing by Ludovico Vicentino degli Arrighi, in a wood block printed book published in 1522. Apple said, well, okay, we can do character substitution now, and technically we could produce a page like this, but we need you to design a font that would enable us to do that.

Y: So what's your design procedure?

C: The first thing we did was go back to our calligraphic teaching manual from Lloyd Reynolds, who was our calligraphy teacher at Reed College, in Oregon. Kris and I studied with him, at different times. Steve Jobs took calligraphy courses at Reed, too. Here's a picture (fig. 22) of Reynolds, standing outside his calligraphy studio at Reed College in 1967 and a sample of his italic handwriting. After the commencement, he printed it out so all of his calligraphy students could have a copy of it.

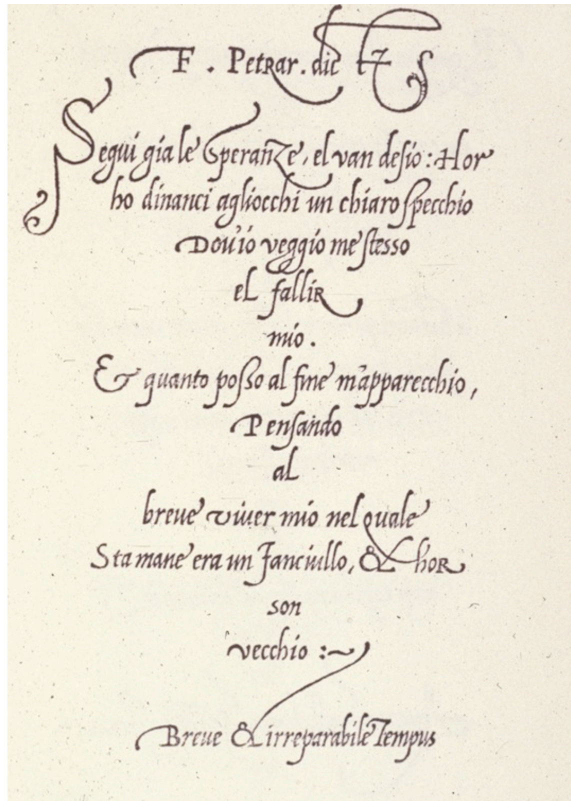


Figure 21: Chancery cursive writing by Arrighi (Ludovico Vincentino), wood block print in 1522. Apple wanted to be able to do this on their screens.

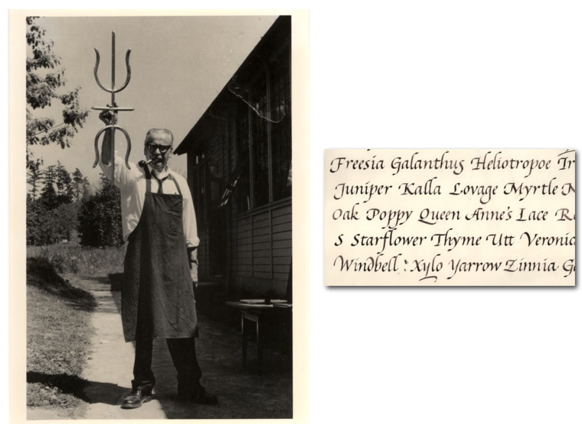


Figure 22: Lloyd Reynolds, calligraphy teacher of Kris Holmes and Charles Bigelow. Standing outside his calligraphy studio at Reed College, circa 1967. A sample of his italic handwriting.

Y: How's your reproduction process based on his teaching manual?

C: Arrighi's manual is clear and elegant, with many fine flourishes, but the letters were cut in wood and are a little more angular than examples of his actual

Figure 23: Variations of 'k'.

Lucida Calligraphy

Apple Chancery

Lucida Calligraphy

Apple Chancery

Figure 24: Comparison of Lucida Calligraphy to Apple Chancery; both designs are chancery cursives by Bigelow & Holmes. Apple Chancery is more like the form of calligraphy taught by Lloyd Reynolds, based on Arrighi's models. The top pair are both set at a body size of 28 pt; in the lower pair, the Apple Chancery size has been increased to equalize x-heights.

handwriting and of other scribes of that era, so Kris wrote all the characters with a pen and worked out as many variants of every letter of the alphabet as she could dream up. For example, if you look at the lower case 'k', there's a very simple 'k', a more complicated 'k', a 'k' that would go at the beginning of the line, a 'k' that would go at the end of the line, and so on. (See fig. 23.)

Kris created her samples based on Reynolds' teaching and manual, and we enlarged them, and then we redrew them. And we made a few changes to make them sturdier looking for typographic use so the hairlines were thickened up a little bit and the characters were made a little wider than they would be just with a pen written character. The result was Apple Chancery. A "chancery" was a medieval or Renaissance clerical office where scribes wrote the documents needed to organize a kingdom or city or organization. A special kind of handwriting used in Italian Renaissance chanceries came to be called "chancery cursive". So, we suggested that this italic handwriting, designed for Apple, could be called "Apple Chancery".

Y: It looks like Lucida Calligraphy.

C: Yes, both were based on our study of italic handwriting with Reynolds, who based his teaching on calligraphers like Alfred Fairbank, who based his on the works of Arrighi and other Italian calligraphers of the 16th century. Lucida Calligraphy has a big

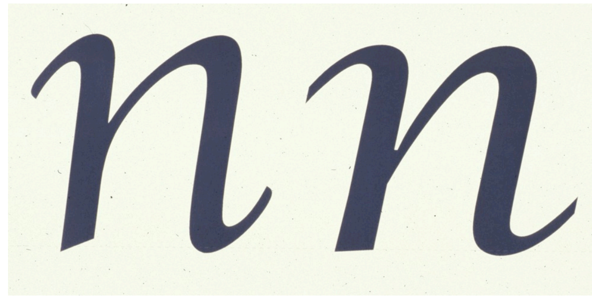
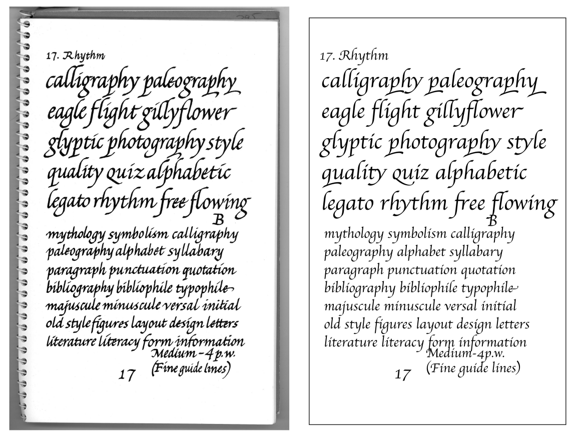


Figure 25: Apple Chancery (left) compared to Lucida Calligraphy (right).



17. Rhythm
calligraphy paleography
eagle flight gillyflower
glyptic photography style
quality quiz alphabetic
legato rhythm free flowing
mythology symbolism calligraphy
paleography alphabet syllabary
paragraph punctuation quotation
bibliography bibliophile typophile
majuscule minuscule versal initial
old style figures layout design letters
literature literacy form information
Medium – 4 p.w.
(Fine guide lines)
17

Figure 26: Reynolds' hand-drawn calligraphy compared to Apple Chancery.

x-height, much bigger than the Italian Renaissance models, so it can align with other Lucida fonts. Apple Chancery stands alone, so it has a smaller x-height, more like the traditional chancery handwriting and fonts of the Renaissance. Here (fig. 24) is the comparison of Lucida Calligraphy (big x-height) to Apple Chancery (small x-height), at the same body size. The type with the smaller x-height looks smaller. But when we designed Lucida Calligraphy, the old Canon printing technology tended to increase dark shapes and some of the details would clog up, like the 'n' here. We made modifications to the shapes to prevent this from happening. But in Apple Chancery we didn't need that any more (fig. 25).

Y: So the shape is more beautiful.

C: Apple Chancery is more like the Renaissance proportions of calligraphy taught by Lloyd Reynolds, based on Arrighi's models. We produced a huge character set for this font. In the end it had more than a thousand glyphs in it. This was how Apple Chancery came into being.

Y: So Apple Chancery was the testbed of smart font technology?

C: It was the most extensive use of Apple's TrueType GX font technology in its first release. Apple also used smart technology in other fonts released around the same time, so Apple Chancery wasn't the only pioneering smart font, but it was the most ambitious at that time. Zapfino is a smart font that came later, in OS X, with even more variant characters in a free calligraphic style. When we finished the Apple Chancery project, Kris made this (fig. 26): on the left is a page from Reynolds' calligraphy book. And on the right is the same page duplicated in Apple Chancery. You can see the difference. The typographic forms are a little lighter. They're a little wider, not quite as rich in variation. But we were very pleased with this, because I think that the spirit of Reynolds' calligraphy is in here. Steve Jobs was at Reed a few years after Kris. He also studied calligraphy there, so he was influenced by the same ideas from Lloyd Reynolds, which he described in a commencement speech at Stanford some years ago. So these traditions and interactions fit together in a coincidental but intriguing way. Apple Chancery is intended to honor Lloyd Reynolds' memory, and in a way also commemorates Steve Jobs' experience studying calligraphy.

Y: Given that all recent fonts are moving to incorporate AAT or OpenType features, what's your recent plan for Lucida?

C: Good question. We recently adopted OpenType for a very functional purpose: a new version of Lucida Math for TEX. Almost 20 years ago, we worked with the firm of Y&Y to make a set of Lucida Math fonts in PostScript Type 1 format for TEX. Berthold and Blenda Horn did a lot of work to make Lucida fonts compatible with TEX. Since then, the Unicode standard has added several blocks of math symbols and alphabets, and OpenType enables glyph substitution. To upgrade Lucida to OpenType, we added more math symbol sets, a new math script alphabet, plus Greek and Cyrillic alphabets, and we encoded all the characters in Unicode. Previously, we offered basic text fonts plus TEX-oriented math fonts like "Math Italic", "Math Symbol" and "Math Extension" for TEX. Those are now combined into one math font in OpenType (`http://tug.org/lucida`). Karl Berry coordinated the project on behalf of TUG, we designed the new glyphs and Khaled Hosny combined the new character sets with the older ones and built the fonts in OpenType format. Several people from the TEX community helped test and critique the fonts. Mojca Miklavec, Hans Hagen, Ulrik Vieth, Will Robertson, Michael Sharpe, Taco Hoekwater, Bogusław Jackowski, and Barbara Beeton. I hope I got all their names right. An international undertaking.

The new Unicode standards for math symbols incorporate style variations as semantic variations. As one small example, in addition to the usual text versions of 'a', we provide separate math versions for upright 'a' and italic 'a', as well as sans-serif and bold variations, which have different semantic meanings in math.

Y: Yes. Because in math equations, upright is used to mark labels, while italic is for variables. Bold marks are used for vectors.

C: So now in Lucida Math OpenType, we include all these variations that are specified in Unicode. Now there are more than 3100 math glyphs in Lucida Bright Math and around 1700 in Lucida Bright Demibold Math.

Y: Amazing. So you are using the new OpenType MATH table feature introduced in Microsoft Word 2007?

C: Yes, but we didn't make the math tables, Khaled Hosny did them. First, Kris and I designed the glyphs, using various tools, old and new, including Ikarus, Illustrator, and FontLab, and then Khaled Hosny assembled the fonts and generated the MATH tables using FontForge.

Y: The latest TEX engines like XeTEX and LuaTEX fully support OpenType, so it's much easier to use them.

C: Yes, that's why TUG suggested we make the upgrade. We also took the opportunity to redesign the math operators. When Donald Knuth designed TEX and his Computer Modern typeface, he used relatively large operators compared to the alphabetic characters. I think perhaps as a mathematician he thought the operator relationships were more important than the variables themselves. But, when we first designed math characters for Lucida in the early 1980s, we made the operators relatively small because we were thinking that the symbols should be proportioned like the alphabetic characters, and that it would be helpful if most of the operators were the same width, either like figures or some other set width, so the symbols could easily be used in tables. Later, we agreed more with Knuth's practice, so we increased operator symbol sizes for the Y&Y Lucida Math fonts. And, after more years of experience working on math fonts and seeing them used by mathematicians and computer science, we believed that Knuth had been right all along, so we increased the sizes of the operators again when making the OpenType Lucida Bright math fonts. Now they are close to the proportions Knuth chose more than 30 years ago.

Y: But I still like the original flavor. Maybe you can leave this as an option for users?

C: We kept some of the smaller symbols as alternates in the fonts for those who preferred them.

The older operator designs are also in the PostScript Type 1 Lucida math fonts, which are still available from TUG, so they aren't lost.

Y: You mentioned that there is a demibold version of Lucida Bright math.

C: Yes, when we were working with Y&Y years ago, we added bold operators because Y&Y and some of their customers said, "We need bold for the symbols as well!" because bold is a semantic category for math variables; logically, bold could apply to operators, too, though currently, not all operators have a semantically bold form. So, for Lucida Math OpenType, we made a whole math font in Demibold. Not only because bold characters have semantic meanings, but because mathematicians and technical authors are logical — they think, if we have bold letters, bold greek, bold scripts, and so on, why don't we have bold symbols? Because mathematicians keep thinking of new ideas and need new symbols to represent them, they keep making little bits of new work for type designers. We try to keep up, but math fonts are never really finished, because mathematics keeps expanding. It isn't clear which math characters really need to be bold, so Lucida Bright Demibold Math doesn't offer bold versions of all the characters in the normal weight. We added a bold typewriter to the Lucida Demibold Math font, because there is a Math Typewriter alphabet in the normal weight Lucida Math, and the same for a bold script, in two styles, chancery and English roundhand, and bold arrows as well as bold operators. I expect we will get requests for more bold characters.

Y: Do you make the symbols bold by hand or by using software to automatically make it bolder?

C: Design by hand. We use software to input shapes and edit contours, but not to make automatic bolds. Some math symbols are easy to embolden because their geometry is simple and clear, but some take a little more work, though most are not as difficult as emboldening alphabetic characters. Generally, you get a sharper, crisper, better design by emboldening by hand because you see what you are doing. For outline font formats, there is no single algorithm to make good bold weights, though I have seen that the algorithms are getting better at making pseudo-bold weights. Early methods made a smeary mess of the shapes. In Metafont, however, there is an easy way to embolden shapes, because Metafont can use a pen metaphor: a nib of a certain size, shape, and orientation follows a path and the image of the character is the trace swept out by the nib. You can keep the same path but make the nib bigger to make the shape bolder. More subtle methods change the size, shape, and orientation of the nib. An outline format like TrueType does not use that metaphor. Because of its pen metaphor, Metafont is closer to traditional writing than to traditional metal typography, which used carved outlines.

For bold characters, a design challenge is, how to prevent acute angled joins, like where the hairline of the arch meets the stem in an 'n', from clogging up when printed with heavy ink or toner? In the first versions of Lucida for 300 dot-per-inch printers, we opened up more white space in those areas, but as printers improved, we removed the cut-outs. On screen, there can be the opposite problem — the backlighted background can make characters look lighter. Digital technology keeps presenting new challenges for designers.

Y: Can the Lucida Math characters be used without TeX?

C: Yes, the characters are encoded in the fonts with Unicode. Applications that let users find characters by Unicode code point or that show the whole glyph set let users access the characters. Equations may not look quite the same when Lucida fonts are used with Microsoft Word's math engine, because the Lucida Bright Math fonts don't have exactly the same metrics nor all the same characters, as Microsoft's Cambria Math font. Our goal was to augment the TeX-friendly Lucida Math fonts for OpenType, not emulate Microsoft's font, but we always enjoy designing new characters, so if user feedback tells us that we should include the Cambria math set as a subset of Lucida Bright Math, I expect we will eventually include those characters.

Y: You mentioned chancery script and roundhand script. Are both included, and what is the difference?

C: Originally, Y&Y used our chancery script, Lucida Calligraphy for the default math script. A chancery script for math is found also in Herman Zapf's script capitals in the Euler fonts. A chancery script is "cursive", which means a fast, "running" style, but the letters usually don't join. Some TeX users asked for the English style of roundhand script, which is more common in math composition than the chancery style. In English roundhand, the lower-case letters join, and there is a strong difference between thick stem and thin hairline strokes. Kris designed a true English roundhand face, based on her studies of English writing masters. That is now the default in the Lucida Bright OT math fonts, but the Lucida chancery characters are still in the font as alternates for those who prefer them. Both the chancery and roundhand scripts have normal and bold weights. A very different set of capitals in Lucida Math is the Blackboard Bold set, in which the forms are based more on geometry than handwriting, but they are not purely Euclidean constructions — the capitals relate to the Lucida Sans capitals. We also made a bold Blackboard Bold for Lucida Bright Math demibold.

4 Metafont and the TeX world

Y: You mentioned Metafont's pen metaphor. Do you think the idea of the pen is still useful in the design of a font?

C: Yes, the *idea* of the pen is still powerful, but the long history of metal typography firmly established the outline metaphor. Whenever a type designer, called a punch-cutter until the end of the 19th century, tried to imitate a handwriting style, he had to cut it in steel. Also, the letters had to be cast separately, and for economy and efficiency, there could not be many variant letters or ligatures — characters tied together. Probably the greatest punch-cutter of scripts of all time was a 16th century Frenchman named Robert Granjon. He cut many different fonts of roman, italic, chancery, and cursive blackletter, as well as Armenian, Cyrillic, Syriac, Arabic and other non-Latin scripts. So, punch-cutting could imitate handwriting in the hands of a master. It is much easier to create type today; it doesn't have to be laboriously cut in steel, but even now, most typographic scripts are created as outlines.

In early digital typography, companies were in a hurry to reproduce metal or photo type in digital form. Helvetica, Times Roman, etc. Even Lucida, an original design for digital, was based on an outline metaphor. But, at Stanford, Knuth explored the pen metaphor in his own creation of Computer Modern, and also commissioned the Euler font designs from Hermann Zapf.

I should say a little about how the Euler fonts were produced in Metafont. Zapf drew the letters as outlines; after a career of four decades, he knew well how to render handwriting in outline drawings. The reverse process was much harder for those of us working for and with Knuth at Stanford: how to turn drawn outlines into pen-based paths in the Metafont metaphor. Knuth himself could have done it, but he was busy finishing TeX and Metafont, so he assigned the project to one of his talented graduate students, but progress was slow, so then my students also became engaged in the project, and yet it still went very slowly. The students despaired of ever getting true Metafont characters to match Zapf's drawings. Eventually, I advised them to give up on the "meta" aspect and the pen metaphor, and instead digitize Zapf's drawings as outlines, using a simple-minded hack: set the Metafont pen nib to be very small — one pixel — on a high resolution field, and code the outline contours as paths. This worked well, the characters matched Zapf's drawings, and the production went much faster. However, the resulting characters were not "meta". Normal weight could not be turned into bold by changing pen nibs, serifs could not be altered by changing nibs. Outline representations of characters are basically unintelligent blobs, whereas Metafont representations have structure, but we were not able to reconcile these two different approaches.

I regretted that neither I nor my students could see how to solve the more difficult problems. Given a shape traced by a pen or brush, we can digitize the graphical trace in various ways, but given a drawn outline, it isn't at all clear what path and what pen produced that shape, nor even if that shape can be made by a pen and a path. How to make the characters "meta" — that is, how to design them so that bold, narrow, and other variations can be produced by substituting virtual pen nibs — adds another layer of difficulty. I've always felt guilty about turning an intellectually fascinating but very difficult problem into a simpler but achievable solution under constrained circumstances. Nevertheless, there were practical advantages to the outline solution. After the Euler fonts were produced as digital outlines with Metafont, a group of mathematicians and programmers were able to translate them into the PostScript Type 1 format: Berthold Horn at Y&Y, Henry Pinkham and Ian Morrison at Projective Solutions, and Douglas Henderson at Blue Sky Research. A few years ago, the Euler fonts were revised with further contributions by Zapf [4].

I should mention that the Euler project at Stanford was using Metafont79, not the current Metafont(84). In `mf79`, only the pen metaphor was available; in `mf84`, outlines are also directly supported. Indeed, it was partly because of the Euler experience that Knuth completely rewrote Metafont to support outlines as the primary drawing mechanism.

The pen metaphor is still valid as inspiration, but it has mostly been ignored in commercial font development. Today, nearly every digital font designer uses a visual application like FontLab or Fontforge, not Metafont. So I don't think there is any need to use the pen metaphor for actual production. Even to capture handwriting, as long as the shapes produced by pen strokes can be turned into outlines, designers are happy about it. I am sometimes sorry to see that the spirit and grace of the moving hand and tool, whether pen, brush, or reed, are lost in modern typographic technology, but now that the basic problems of outline font technology are solved, perhaps someone in the future will work on restoring the human action.

Y: So you think Metafont is too hard for designers.

C: Yes, at least for visually oriented designers. Metafont is mathematically based, whereas most designers rely on their visual intuition and avoid mathematics. Metafont uses an abstruse programming language to describe characters, which must be written and tested like computer code, and which makes it nearly impossible for visually trained designers to learn to use it. The intersection of Programming Experts and Design Experts is nearly the Empty Set, though some younger designers both write code and create typefaces, but in the outline metaphor. If Knuth had developed a more user-friendly interface to Metafont, or if someone else had successfully worked on a project to automatically record a real pen or brush movement and determine the virtual pen that produced the resulting shape, I think the pen metaphor would have been more widely adopted. Remember, too, that at the output end, all the font engines for screens and printers were biased toward outlines, beginning with PostScript, and followed by TrueType. Nevertheless, a fair number of fonts have been produced with Metafont, especially for non-Latin alphabets and character sets, symbol sets, and others.

Y: I think the same for TEX too.

C: TEX fits in with a technical, logical intellect. For visual designers who prefer WYSIWYG interfaces, TEX is difficult. Very few graphic designers or typographers appreciate it. However, that is not true for the thousands of mathematicians, physicists, computer scientists, and others who use TEX to write scientific and engineering papers.

A personal anecdote to support that claim: My neighbor is a retired mathematician, Norman Alling. He wrote a book on real elliptic curves and taught himself TEX in his 50s, so he could compose his book and papers himself, and he still uses TEX now, in his 80s. He says TEX liberated math journals and authors from dependence on commercial math typesetting, which was slow, expensive, and fraught with typographical errors needing proofreading and correction. When I told him that some people suggested that Knuth could have better spent his time finishing the Art of Computer Programming books instead of spending a decade developing TEX, he replied: Oh no, TEX liberated so many mathematicians and scientists from the bottleneck of typesetting that it was a great boon to all of math and science, more important for the world-wide science and technical professions than Knuth's unpublished books on computing, however excellent they might be. That's just one opinion, of course, but it suggests how liberating TEX was and still is. And, of course, Don Knuth is still working on his books, and many people hope for his success in finishing them.

As a side note, did you know that Knuth's work on typography was anticipated by the Italian mathematician Giuseppe Peano, a founder of mathematical logic? Peano was concerned with the precise forms of mathematical notation in print, and was frustrated by the difficulties of getting mathematics typeset and printed, so to further his grand ambition to publish an encyclopedia of mathematical formulae, he purchased his own printing shop and took classes in composition and printing. Typography is the graphic art that seems to appeal most to mathematicians (apart, perhaps, from the prints of M. C. Escher). Do you know the mathematician who developed the Unicode TEX fork for non-Latin scripts? He also wrote the book *Fonts & Encodings* [5].

Y: Oh, you mean Yannis Haralambous's Omega?

C: Yes. His book is a massive volume of information. It touches on nearly every subject in digital typography, often in great detail. His Omega system deals with non-Latin typography, which Kris Holmes and I also find fascinating, but we look at it from the character design aspect, not the programming aspect. Haralambous developed an actual system. It's really an impressive body of work.

Y: Yes. And the source code has been merged into the future version of TEX called LuaTEX.

C: It's wonderful how many dedicated people continue to contribute to the expansion of TEX.

Among computer scientists who showed early interest in digital typography were the developers of Unix at Bell Labs. In 1979, Ken Thompson, Brian Kernighan, Joe Condon, and perhaps others, wrote software for Unix systems to drive the Linotron 202, a new digital typesetter that Bell Labs had bought. They found that the Linotron 202's factory-installed software was buggy and that the font encryption prevented them from inputting their own graphics. In a brilliant summer project, amusingly and succinctly described in an internal Bell Labs report, they disassembled the typesetter's own operating system and replaced it with their own software. They also decrypted the typesetter's font encryption scheme so they could input their own graphics. As well as being less buggy, their new software was faster at processing mathematical texts, although slower at processing newspaper texts — the Labs published technical papers, not newspapers. They also developed software to input digital graphics like diagrams, chess pieces, logos, and so on. The Labs' internal report was a nice description of problem-solving by intelligence. They didn't use big brute force number-crunching to decrypt the machine's software, but simply studied and analyzed its workings, then experimented with their own code. Looking back, it is clear that they were seeing the future, six years before PostScript printers and imagesetters revolutionized digital text and graphics imaging. Their approach could have been more widely exploited, but I believe the paper was not published and their software not distributed with Unix because of legal issues with reverse-engineering. However, their paper has finally been scanned and released for its historical interest; it's on Brian Kernighan's page on the Bell Labs site: `http://www.cs.bell-labs.com/cm/cs/who/bwk/202.pdf`. A modern revival is being reprogrammed by David Brailsford for the Document Engineering 2013 conference.

5 Beyond Latin alphabets

Y: The Lucida Grande fonts in Mac OS X have several non-Latin alphabets, like Greek, Cyrillic, Hebrew, Thai, and Arabic. Do they use advanced typography as well?

C: Yes, to some extent, but not a lot. Modern Latin, Cyrillic, and Greek fonts don't really need advanced typography like Apple's AAT or OpenType. Latin fonts may benefit from the aesthetic possibilities of glyph substitution, but they don't *need* it for legible text. In metal typography, simplification of character sets made typesetting and printing more economical, because fewer characters needed to be cut, cast, stored, and composed. Hence, by the middle of the 16th century, most abbreviations, ligatures, and variant forms of characters had been eliminated from standard roman and italic fonts. For Cyrillic type, a similar simplification took place in the early 18th century under the direction of Czar Peter the Great of Russia. Greek fonts, which could be very complex because of many ligatured forms and complex sets of accented vowels, were gradually simplified over the centuries by elimination of ligatures and variants. In the late 20th century Greek "monotonic" standard, ligatures are eliminated and the number of accented letters greatly reduced.

Typefaces based on cursive handwriting, however, tend to have more joining forms and context-sensitive complexity. Apple Chancery has more than 1,000 characters, with hundreds of variants and ligatures. Herman Zapf's Zapfino has more than 1400 glyphs, including letter variants and ligatured forms using advanced typographic features.

However, the Arabic writing system really needs advanced typographic support in order to make traditional styles practical in typesetting. The first release of Lucida Grande Arabic in 2001, a sans-serif design but in the Arabic Naskh style, definitely made use of advanced typography, in the form of Apple's AAT system. Most Arabic typefaces today use OpenType, and many interesting and elegant Arabic typefaces have been designed in the past decade, because of the new freedoms of advanced typography and glyph substitution.

The Devanagari writing system used for modern Hindi and some other languages of India, and also for classical Sanskrit, also benefits greatly from advanced typography. For Sun Microsystems we designed a Lucida Sans Devanagari face that uses OpenType, but it was not included in Lucida Grande.

Y: How did you expand Lucida from Latin to other alphabets?

C: Our teacher of calligraphy, Lloyd Reynolds, emphasized that written forms must have life and action. He liked to quote an ancient Chinese art philosopher, Xie He, whose first principle of painting was, "qiyun shengdong", spirit breath rhythm life movement. Nearly all typographic forms were originally imitations of handwriting, though the subsequent evolution of typefaces takes different routes. Because Lucida was based to a large extent on Italian Renaissance handwriting, we tried to base the Lucida Greek alphabet on older Greek handwriting. Kris practiced writing medieval styles of Greek, and then we modernized them into sans-serif styles. Of course, many Greek capital letters are shared with Roman forms, but by starting with handwriting for the lower-case, we tried to give it more life and action. We used similar principles for Cyrillic, though its modern forms are more directly derived from typefaces, not traditional handwriting.

Again, for Hebrew and Arabic, we first studied traditional calligraphy. Arabic writing has a long tradition of elegant calligraphy, but it is difficult to distill that to fonts that are legible in small sizes on computer screens. Apple asked that Lucida Grande Arabic look almost as big as Latin at small screen sizes, to be legible in menus, captions on icons, and so on, so we designed it as a sans-serif design in the Naskh style, based on a design we had also done for Sun Microsystems' Java Developer Kit. Lucida Grande looked very legible at small sizes, and was shown in a book about Arabic typography by Huda Smitshuijzen AbiFares [1]. Later, however, some people told Apple it looked too big when printed, so Apple replaced it with a more traditional looking Arabic font as default. However, after Lucida Sans Arabic for Java and Lucida Grande Arabic, several new sans-serif Arabics have been designed with big "looks", so the design idea has become popular.

Our first international font was Lucida Sans Unicode for Microsoft, in 1993. It was one of the first

TrueType fonts to incorporate several different alphabets — Latin, Greek, Cyrillic, and Hebrew, plus mathematical and technical symbol sets. We wrote an academic paper about how and why we did it [2].

Apple asked to include that kind of international Lucida in an operating system that never came out. It only had a code name but was never released.

Y: It's called Copland.

C: Yes, Copland. So Apple acquired a license for Lucida Sans. At that time it was not called "Lucida Grande", but when we included more glyphs for Latin-based orthographies, including Turkish, Czech, Slovak, and many others, plus Greek, Cyrillic, Arabic, Thai, and other international languages, it became much grander, so Apple thought it should be called "Grande" to emphasize its larger, more grandiose character repertoire.

Y: So in 1999 if Apple wanted to use something new, why use Lucida? It was already 15 years old. Why not ask you to create something new for the 21st century?

C: Great question. I think it would have been a good idea to do something totally new. We love to do new designs, but Apple didn't have time for the development and testing of a totally new font. A fact about text fonts is that it takes most of them years to prove themselves. Ornamental faces can become quick "hits", but text fonts are usually slow to become popular. Continuous reading is a subtle process and preference for fonts emerges slowly. Adobe considered Lucida in the very early days of PostScript, 1983–84, but Lucida had not yet been released and Adobe was unsure about whether it would be popular on the Apple LaserWriter, so they chose well known existing fonts. By 1999, Lucida was well known and proven in practice, so Apple wasn't taking a risk by making it the system font for OS X. It was already well liked by computer users.

Y: Speaking of Mac OS X, why there is no italic variant in the Lucida Grande typeface?

C: Oh, interesting question. There are true italics for Lucida Grande, but Apple did not release them with OS X. Next year (2013), we plan to release them ourselves. Lucida Sans Italic is a cursive design, based on the same Arrighi chancery handwriting of the 16th century that inspired Lucida Calligraphy and Apple Chancery, but we simplified it greatly for the sans-serif style. Eric Gill first did this with his Gill Sans Italic in 1928, and Hans Meier's Syntax italic of 1968 is also a cursive design, though he kept the humanist form of lower-case 'a' and 'g'. Lucida Sans Italic was first released in 1985. In the decades since then, sans-serif italics have become a style popular with several designers. For example, in Frutiger Next, released in the year 2000, a true cursive italic replaced the original slanted roman of 1976. I think it is a sign that sans-serif is continuing to mature and evolve.

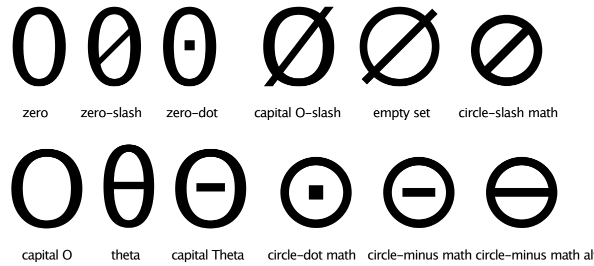


Figure 27: Variations on zero, and related forms.

On the computer screens of 10 to 12 years ago, simple forms usually looked better than complex ones, and some designers preferred the simplicity of slanted roman to the more complex cursive of true italic. In a different way, Donald Knuth used both true italic and oblique styles in TeX, for different semantics. Oblique designs are not a new idea, because sloped roman for italic was proposed by Stanley Morison in 1926, and most sans-serifs used slanted romans, not cursive italics. In our Lucida Math fonts, we provide both true italics and obliques.

Y: Here's a question from a friend. Why in Lucida Grande is the en dash actually shorter than the hyphen?

C: Your friend has spotted an interesting problem. In Lucida Grande, the default hyphen is not a true hyphen but a hybrid between hyphen and minus sign from the ASCII standard. It is longer than a true hyphen but shorter than a true minus, because people use it for both functions. The en dash is by definition one-half of an em square wide, including a little space on each side. The minus is wider than an en dash, because Lucida math symbols are fairly wide. So, when we made the hybrid hyphen-minus, it turned out wider than the en dash. I hope to adjust the disparity in the next version of Lucida Grande when we release it ourselves. Currently, it is only distributed by Apple. (Has your friend spotted any other anomalies that need fixing? Now is a good time to ask about them.)

Another interesting problem that affects technical users is the design of the zero (fig. 27). If you peek into the unencoded glyphs of Lucida Grande, you can find alternate forms of zero. The encoded form is the standard open or empty zero, nothing inside. It's a nice iconic symbol, like an empty set. However, in computing, the problem of confusion between zero and capital 'O' has been debated since the 1960s, e.g., in the journal of the ACM (Association for Computing Machinery). So, some computer fonts have a zero with a slash. OK in English, but in Danish and Norwegian, there is a slashed capital 'O' and a slashed lower-case 'o', which can be confused with the slashed zero. Other people prefer a zero with a dot in the center, but that can be confused with Greek capital Theta, and programmer friends of mine say it is aesthetically displeasing and call it the "fly-speck zero". For Apple, in Lucida Grande, we provided all three variants, and recommended that Apple could use whichever one seemed correct for any given localization. The open zero is the default. When Apple asked us to redesign Monaco for TrueType and System 7, we made the zero with a slash because programmers were using the font. We also differentiated the capital 'I', lower-case 'l', and figure '1', for programmers. But, some people still don't like the slashed zero. In Lucida Console. we used the open zero because there it is less likely that zero and capital 'O' will be confused, because the 'O' is shorter than the zero. But, in the next version of Lucida Console, we will use a slashed zero.

The more characters in a font, the more design puzzles and potential conflicts between design, culture, and technology we encounter. Here's another little example, but it takes a while to explain. When we made Lucida Console for Microsoft, we were asked to include the Unicode character 010F, called the "dcaron" or "Latin small letter d with caron" (ď), which is used in Czech and Slovak, two related Slavic languages of Central Europe. The d-caron marks a phonetic variant of the sound represented by the letter 'd'. In early Czech orthography, it was a little dot above the letter, and that eventually became an inverted circumflex, called a "haček" or "caron" in English. The capital form is a 'D' with haček above it, but in printing, the lower-case became a 'd' followed by an apostrophe, probably because that was easier to make in metal type. It is difficult to fit a haček above the ascender of the 'd' in metal type. In a fixed-pitch font like Lucida Console, the apostrophe variant is difficult to design because the apostrophe takes up space to the right of the letter. A designer can squash the width of the 'd' and cram in the apostrophe, which I don't like, or fit a caron over the bowl of the 'd' but not above it, but some Czech readers don't like that. For Lucida Console, we couldn't hang or kern the apostrophe beyond the right edge of the fixed-width cell, nor put it above the ascender, because those violated a screen display rule in Windows NT, and in any case could overlap a neighboring letter. So, we made three different versions and put them in the font we delivered to

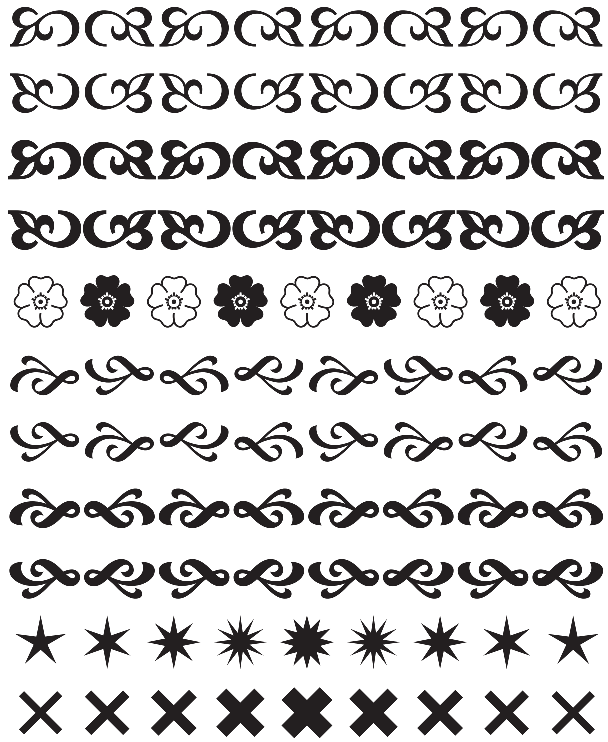


Figure 28: Wingdings fleurons (originally for Lucida). From top, the rosebud fleurons have four symmetry variants in normal and bold weights. The eglantines (wild rose flowers) have outline and filled variants. The vine tendrils have four symmetry variants in normal and bold. The stars have a range of points and the Xs have a range of weights.

Microsoft, suggesting that Microsoft ask their Czech localization experts which one is best, and use that in the standard Unicode encoding. The font got released with the caron positioned above the bowl of the 'd', and no one complained, until several years later, Microsoft told us that some Czech users, probably programmers, because they are the main users of Lucida Console, didn't like that default version. Microsoft asked us to fix it, and we said, sure, it's easy. Just replace it with the alternate character from the original font we delivered. That was maybe 10 years ago, but I don't think it has been replaced yet. We will release a new version next year, with the preferred version, now that we know it is preferred.

6 From the present to the future

Y: I also have a question related to Microsoft. What was in your mind when designing the Wingdings typeface? Is there any connection between the symbols and their letter representation?

C: No. It is a complicated but instructive story. The characters in the Wingdings font originally came from three fonts of non-alphabetic characters called Lucida Symbols, Lucida Stars, Lucida Arrows, and Lucida Icons. We designed them to work with Lucida, and to be useful or decorative, or both. There are several pretty ornamental “fleurons” or flowers in Wingdings, in addition to more functional designs (see fig. 28). Microsoft licensed and distributed them in a beta-test release with Windows 3.1 in 1991. Then Microsoft bought outright the icons, arrows, and stars fonts, to make a new, exclusive symbol font for Windows. The other Lucida fonts were licensed, so B&H still owns them. Back then, Microsoft Windows was distributed on floppy disks, and Microsoft found if they included all three symbol fonts, they would need an extra floppy to hold the files, so they decided to merge only their favorite symbols from the three fonts into one font.

Y: But floppies are cheap.

C: Yes, but since there would be tens of millions of copies of Windows sold, it would have cost Microsoft a lot more money. Another issue in those days was that symbol fonts had to be mapped to the keyboard for characters to be accessed. One symbol was mapped to capital ‘A’, another to ‘B’, and so on. Some Microsoft managers and font advisers chose their favorite symbols from the three fonts and had them merged them into one font. This merging meant that all the original mappings from the B&H fonts were changed by Microsoft. This font became Wingdings.

Y: But then users found interesting sequences.

C: Yes. The first discovery was that the character sequence “NYC” in Wingdings was rendered as a skull and crossbones, a Star of David, and a thumbs up gesture. This was interpreted as an anti-semitic message, in an article in the New York Post newspaper, but the popular magician, Penn Jillette, wrote a column for a computer magazine debunking the Post’s story, pointing out that the assignment of symbols to letters on a keyboard inevitably results in sequences that can be interpreted as meaningful, even when no meaning was intended. The problem was that Microsoft hadn’t sent the newly encoded Wingdings font out for beta-testing. From a technical engineering view, the font worked perfectly. It was in user psychology that the problem arose. This is why software should be tested outside a firm. Later, we were told about many other messages supposedly found in Wingdings. One was that the symbols assigned to the sequence “LBJ JFK” proved that Lyndon Johnson was complicit in the assassination of John F. Kennedy.

Y: And when 9/11 happened, you became the most reported type designer in history.

C: Yes. An email went viral on the Internet, claiming that in Wingdings, typing ‘Q33 NY’, supposedly the flight number of the first plane to hit the Twin Towers, would show an icon sequence of a plane flying into two towers, followed by the skull and crossbones symbol and the Star of David. But, the real flight number wasn’t Q33 NY. Somebody just made it up. In the Wingdings design, the rectangles were icons for documents with text, not buildings. And the font was made 10 years before 9/11. But none of that mattered to gullible journalists who didn’t check the supposed “facts” they read on the net, and asked me after September 11, 2001, “Why is Wingdings associated with the terrorists?” This was back when some naive journalists still believed what they read on the web.

Y: [Laugh.]

C: I also heard that people were typing the names of their husbands and wives in Wingdings, to find clues to whether their spouses were “cheating” on them. People like to find hidden messages, even when the messages are noise, not signal. There is a whole field of study about why people like to believe in hidden messages and conspiracy theories.

Y: So Wingdings became a hot topic at that time. Many people talked about it. Has there been an increase of public awareness and interest in typography in general over the years?

C: Yes, very much. People talk and write much more about these issues now. About legibility. About whether some typefaces make a text more believable, some less believable. 30 or 40 years ago such discussion only appeared in design journals. But now I see discussions of fonts every week in newspapers like the NY Times or magazines, and of course on the Internet. I recently saw a book of Guatemalan poetry entitled ‘Times New Roman Punto Doce” (Times New Roman 12 point).

People also react strongly to typefaces used in movies. In the movie “Avatar”, the font Papyrus was used in subtitles for the Na’vi language of the alien people, probably because it has a charming, rough, hand-made look, but the movie-makers didn’t think about the reactions of font-familiar viewers. Instead of enhancing the experience, it distracted viewers from the story: young people felt cheated because they recognized Papyrus as an Earthly typeface bundled with millions of personal computers. My students made complaints like, “This is not a new font from outer space! We’ve already seen it in Mac OS X!” It caused a lot of comments on the web, most of them negative.

Y: Haha, interesting.

C: What’s interesting to me is not the specific opinions, which on the Internet are often either love or hate, but that so many people voiced their opinions. Here, I should make an appeal, as some of the commenters on the web also did, for the type design profession. James Cameron spent $300 million dollars making “Avatar”. He even hired a linguist to invent the spoken language of the alien people, the Na’vi. So, instead of using a common font found on millions of computers, he should have commissioned a young, or old, type designer to create a totally new, unique typeface for those subtitles.

Speaking of subtitles, I enjoy watching Chinese movies but have to read English subtitles to understand the dialogue. I understand that in China, movies are also subtitled, so speakers of different dialects of Chinese can understand what is being spoken. In the American release of “Crouching Tiger, Hidden Dragon”, the English subtitles used Lucida Sans Italic. An exciting HK-crime movie was “Infernal Affairs”. The Chinese poster is typographically intriguing because the design of the characters is like a maze and suggests the complexity of the story, and the title means “Endless Path”; a nice integration of visual form and symbolic meaning. Because “dou” is “tao”, the movie is a Buddhist and Taoist lesson :-).

Y: Yes. And at the same time, movies promote typefaces too. I really love the 2007 film “Helvetica” directed by Gary Hustwit. Maybe sometime in the future we can make a film about Lucida.

C: Yes, Helvetica is a good movie that reveals a lot about why people like type. And since you mention the idea, I should say that a movie is now being made about Kris Holmes and her work. It will include Lucida, and other things. So, we can hope that will someday be shown on the big screen, too.

Ah. Another example of types and personalities is in the presidential elections of 2008 and 2012; both sides cared very much about the typefaces they used.

Y: In 2008, Obama used Gotham, which is also used in Batman.

C: Yes, Gotham is an urban sans-serif, while Optima, a more delicate semi-sans-serif was used by the 2008 John McCain campaign. The public analyzes the typeface to tell the personality of the candidates. The International Herald Tribune praised the Obama choice for its “potent, if unspoken, combination of contemporary sophistication with nostalgia for America’s past and a sense of duty.” Wow! In the 2012 election, many of the Romney signs used Trajan, which is a modern revival of lettering used on imperial Roman inscriptions. I wonder if the public got the idea that Romney’s ideas were 2,000 years old, or that he wanted to be an Emperor. Certainly, the visual impression of Trajan is formal and stiff, like Romney.

Y: In the election happening just now in Taiwan, this is also true. The Democratic Progressive Party in Taiwan cares about campaign design very much. I really love all the posters, photos and clips they made. It reflects the novelty and neutrality of Tsai Ing-wen, the candidate. In my view, the Democratic Progressive Party does a much better job than the Nationalist Party of China. But it’s a pity that typography does not mean everything. Today, this morning they lost the election. But it’s interesting to see that Asians are following closely.

C: Yes. I don’t think typefaces can influence elections much, unless the candidates are otherwise indistinguishable and one uses Comic Sans and the other, Times Roman. But speaking of elections in Taiwan, digital typography has made it much easier to develop and use expressive typefaces for Chinese. My student, and your friend, Xuan Zhang and I did a study of the expressiveness of Chinese typefaces, but, alas, we didn’t finish the study before he graduated. Nevertheless, I was intrigued to study the wide variations in Chinese type designs available today.

Back to your question about public awareness of typography, there are numerous blogs about typefaces now, and discussion groups like Typophile. People write how they love or hate certain typefaces. There is a site that express how much the blogger hates Comic Sans (Ban Comic Sans). It’s amusing and not too serious, more fun than nasty.

Y: The same for Arial too!

C: I haven’t seen the anti-Arial sites, but I confess, I disapprove of Arial for ethical reasons. I feel it is a too-close imitation of Helvetica, a nearly identical style with the same width metrics, x-height, capital height, stem weights, and proportions so it can replace Helvetica but be just different enough in little details to not be an obvious rip-off or plagiarism. It was said that Monotype offered Arial to Microsoft much less expensively than what Linotype wanted to license Helvetica, so Arial is a font made for business reasons, not for artistic integrity, and as such, it doesn’t advance the art of type design.

Y: Oh, I have another question related to Helvetica. As you know, Apple switched to a Helvetica flavored typeface on the iOS platform and many professional Mac apps as well instead of continuing to use Lucida Grande. What’s the motivation behind that?

C: I don’t know the answer. I guess that Steve Jobs saw the Helvetica movie and decided he wanted to switch to that instead of Lucida. Did you know that Steve Jobs and Kris Holmes studied with the same calligraphy teacher, Robert Palladino, a former monk who taught at Reed College in Oregon? But not at the same time. Typefaces have a lifetime. Text faces get adopted slowly, sometimes over decades, and slowly become old-fashioned, also over decades, though some older designs get revived. Lucida took several years to become widely used in the mainstream computer world, but by the 1990s, it had been adopted by Microsoft and Sun Microsystems, and was licensed by Adobe as well. Apple licensed it for System 7, but then postponed release until the Copland system, and when that was never released, Lucida was postponed again until it became the user interface font for Mac OS X.

Y: Then Apple's Mac OS X has been using Lucida for more than ten years.

C: Exactly. Designers of operating systems sometimes change fonts, just as magazine designers do. Helvetica, in my opinion, was not very legible at low resolutions and small sizes on screens. It is too tightly fitted and the letter shapes are too similar. That's one of the reasons we designed Lucida, with more humanistic letter forms and looser spacing for better legibility on screens. But, as screen resolutions increase, and anti-aliasing techniques improve, we see better and better displays (like Apple's Retina displays), so Helvetica's subtleties can now be rendered more clearly.

Y: And what's your opinion on that?

C: Well, we designed Lucida in the 1980s to be an alternative to Helvetica, so that reveals my opinion. Though the designs have roughly the same x-height and stem weights, the letter spaces between Lucida letters are proportionally greater than in Helvetica, and the Lucida letters are more differentiated in shape. Thus when rendering on a computer screen you will find it much easier to read Lucida than Helvetica. It's easy to demonstrate — see, now you are reading questions in your Mail application which uses Helvetica, so you have to move your head (and eyes) much closer to the computer screen.

Y: Oh! Yes, I never think about that. And there is no way to switch back to Lucida in the Mail application.

C: If you keep that posture for too long, maybe your neck will hurt. No typeface is perfect for every size and reading distance. Digital technology makes it easy to scale type to any size, but human vision puts different constraints on type designs. Among other factors, there is a visual phenomenon called "crowding", which limits how closely objects can be spaced. When objects such as letters are too close to each other, you have trouble recognizing them unless you bring the text closer to your eyes, thus making the images, and their spaces, bigger on the retinas of your eyes. In a practical way, type designers have known this for five hundred years and have adjusted smaller fonts to have more space between letters. If you enlarge a photo of a 6 point font cut in 1550 to the same size as a 12 point font cut at the same time by the same punch-cutter, the 6 point font will be wider and more widely spaced. So, types intended for small sizes on screens should be spaced more widely than types intended for large sizes.

Y: Similar question. You and Kris are also two of the authors of Monaco. Monaco was originally the main console font for Terminal application, as well as the font to display code in their Xcode development tool. But they are migrating away from Monaco.

C: Yes, Actually Monaco is one of the Apple city fonts that were originally bitmaps in the first Macintosh. The bitmap fonts with "city" names were created by Susan Kare, an artist and graphic designer who created many of the interface elements for the Apple Macintosh in the 1980s. Later, she left Apple.

Y: Yes. She came to NeXT with Steve Jobs and served as Creative Director at NeXT.

C: OK, and she still designs icons and other digital images. In 1989, Apple asked Kris and me to make new versions of the bitmap city fonts — New York, Monaco, Geneva, and Chicago — to vectorized form in the TrueType font format that Apple invented. So we did. The new fonts began with Susan Kare's designs, but it was impossible to make them exactly the same in vector format, so we had to change several features and proportions. In Monaco, we had legibility in mind, along with the need to differentiate certain letters for better recognition by programmers and technicians. Characters are distinct, and it is difficult to confuse 0 (figure zero) and O (uppercase O), or 1 (figure one), | (vertical bar), I (uppercase i), and l (lowercase l). We tried to maintain the hint of cursive that was seen in the original 'a' in the bitmap Monaco, but we innovated several other features. We wrote a short paper about the project [3].

Y: But now Apple is switching from Monaco to Menlo. What's the main reason behind this move?

C: I don't know, but I guess that one reason might be that Menlo has a full set of italic and bold weights, whereas Monaco has only roman. Years ago, we offered to expand the Monaco family with bold weights and italics, but Apple never chose to do so. I assume Menlo was named after the Menlo Park city in California. Menlo is also free and open source. It's a revision of open source Bitstream Vera and the open source Deja Vu font family based on Vera. Thus I'm not astonished to see Apple adopt it. It's free and they can modify it as they wish.

Y: And I guess because Menlo came from an open source font, most Linux or other open source operating system programmers are familiar with it. So it makes them happy to switch to develop applications for Mac.

C: Yes, that sounds reasonable.

Y: You just mentioned screen resolution got much higher over these years. Do you think more people will switch to screen reading?

C: Yes. In 2009, less than 3 percent of publishers' book sales were e-books, but today, around 20 percent are e-books. Based on the current adoption rate, I guess that screen publications will outnumber paper ones in 10 years or sooner, including books, magazines, and newspapers. Of course, prediction is difficult, especially about the future. Maybe it will be sooner, and maybe later. In the 1980s, screen resolution was not high enough to render type well, so most people still read newspapers. However, by the 1990s, computer screens got better and better. While for older people, newspapers are still their main reading media by habit and preference, many young people spend more time reading computer screens than reading print newspapers. Now, on this graph, you can see that the readership trend line for print newspapers is dropping quickly as screen resolutions increase. Now that we have very high resolution displays, for example Apple's Retina displays, I expect that in the future more people will read from screens than paper. The trend is accelerating with the iPads from Apple, the Android tablets from various firms, and the Kindles from Amazon. For 500 years, printing on paper was the dominant information technology for Europe and most of the world. Now, digital media are the information technology of the 21st century, but for humans to receive the information, it must be read, and reading requires typefaces and fonts. Happily, most fonts have made the leap from analog to digital. Not all of them work as well in digital, but that provides opportunities for designers to create new typefaces, and to revive and revise older ones for new technology. Print might be dying, but typography is living better than ever.

Y: In the era of LCD screens at max 500 DPI, must the design of a font be influenced by the screen resolution, which is still low compared with the 600 DPI of a common laser desk printer, so that it can be used also in a tablet/ebook reader without many problems?

C: Good question. With high resolution screens, finer details of a typeface can be shown. Hinting has already been abandoned on Apple's Retina displays. Steve Jobs claimed the Retina display has so high a resolution that the human eye cannot see single pixels any more. This is not always true, according to some scientific study, but yes, you can expect that screens and prints will look much more similar these days.

Y: So there is a trend of convergence between digital and print type, as the quality of display improves?

C: Yes. I think so. But there's still something one should care about — for display fonts, a larger x-height and more letterspacing will make the font easier to read on a display. Also, as I have mentioned, the same glyph looks thinner on display than on prints so a font with a little bit of darkness will be better. You should care about these things when designing a typeface.

Y: Even without background illumination, most E-ink readers also choose dark fonts.

C: Yes. This is because the screen's resolution is still not high enough. And moreover there are two other reasons to make the problem worse. First, E-ink has a far coarser gray scale than modern computer screens and doesn't have RGB subpixels. Thus, subpixel rendering doesn't work on those devices, and normal font anti-aliasing works more poorly than on a computer screen. Also, the background color of an E-ink display is already gray. A darker font is more legible.

Y: You just said people will read more and more on a screen. I have a related question. What do you think of the recent hype over web fonts?

C: Oh, yes this is a good thing: web fonts enable readers to read web pages just as books — previously only a limited collection was available for designers to use.

Y: Yes, with the release of the WOFF (Web Open Font Format) specification as an open standard, more and more browsers support it. So designers can use whatever font they like.

C: And moreover, hobbyists are able to create their own fonts and release them to the public to get wider adoption, while previously all of these can only be done by professionals. Web fonts lower the barrier of typographic design and you will see more typefaces appear in the near future.

Y: Then what about the downsides?

C: Well, a web font is much easier to get pirated, and it's harder for type designers to make money from it. That's why TypeKit or similar businesses were born. We should explore this market more to find a reasonable business model for the type designers to make a living.

7 Ancient type digitizations

Y: You said that nowadays hobbyists create their own typefaces, and release them to the public. Last month, font designers in China had a heated argument on one font created by a hobbyist. His name is Digidea. He bought a Kangxi Dictionary which was the standard Chinese dictionary during the 18th and 19th centuries. The dictionary contains 47,037 characters including obscure, variant, rare, and archaic characters. Then he scanned all the characters into the computer, and use auto-tracing tools to trace the outlines of those characters. Finally he released the font called KangXiZiDianTi (see fig. 29).

C: This looks amazing!

Y: Yes, but when you scale the font, you see problems And even if you don't scale the font large enough, you will see uneven thickness among glyphs, even in a single stroke as well.

C: Ah. I see that now. But first of all I should say, if he did not use auto-tracing, this font wouldn't be possible.

Y: Right. It's a huge amount of work — manual font creation would take one person years to do fifty thousand characters.

C: But in most situations you have to. This typeface is lucky, because it's a reproduction of a typeface in a dictionary, where almost all glyphs are presented. The "reviver" of the typeface is lucky to have such a wealth of characters to start with. But in most cases, type designers are not lucky. What if we want a typeface in Xizhi Wang's style? Or Mengfu Zhao's?

Y: We should ask someone who is really good at those styles to write them, or at least we should ask experts to analyze these styles and figure out the underlying logic of the handwriting to provide guidance for the type designers to make glyphs according to these rules.

C: Yes. This is a wonderful challenge involving art, practice, and logic. You write the Slender Gold style devised by Emperor Huizong of Sung. Perhaps you could write a large set of characters at a big size and scan those! And being a computer scientist, you could think about the logic of artistically combining the strokes to make new characters that aren't in any extant examples of Slender Gold. A good hobby for a computer scientist! There was a calligrapher in Japan, Yanagida Taiun, who studied Xizhi Wang's style very well. He practiced Xizhi Wang's Lantingji Xu so well that he could create very good copies, at least, I am not able to distinguish them. I have read that none of Xizhi Wang's original writing survives, only copies, so modern calligraphers make copies of copies. In western calligraphy, there

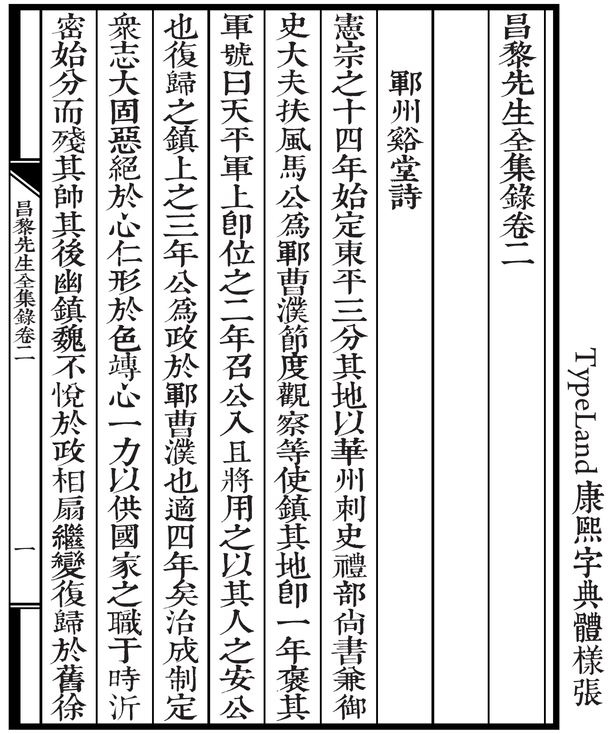


Figure 29: The KangXiZiDianTi font, (re)created by Digidea Lee, 09-26-2010, `http://blog.typeland.com`.

are similar traditions. Many 20th century calligraphers learned the chancery cursive style from a book by Ludovico degli Arrighi, published in 1522. But, the calligraphy was cut in wood blocks, so what some people today are really copying are the wood cuts, not the original handwriting, though some actual samples of Arrighi's handwriting do survive. What Kris Holmes and I and many others learned of chancery cursive was based on modern calligraphers — our teachers — who had reinterpreted how to write chancery from the early printing. We were taught handwriting, not woodblock graphics. For Apple, Kris Holmes designed Apple Chancery, which is an interpretation in digital type of our calligraphy teacher, Lloyd Reynolds, whose handwriting was based on Arrighi's manual and on manuals by English calligraphers who reinterpreted Arrighi.

Y: But in the previous case, if all the glyphs, or perhaps most of the glyphs are available, or those not available can be derived from available parts, do you think ancient type can be made by auto-tracing, or they must be fine tuned by a human?

C: This is a very important question: what is the best way to "revive" a script or typeface from old times? Calligraphers do it by learning to write so their results resemble scripts in surviving old manuscripts. It is like choreography for the hand. You learn a dance of the pen or brush and the traces of your moves are the graphic image of the script. Kris Holmes studied dance, and that is why her scripts are so lively with implied motion. (Lucida Handwriting, Kolibri, Isadora.) We can understand why some of the old calligraphers especially in China and Japan, saw a mystical aspect of calligraphy, influenced by Taoism or Buddhism. Movement, which involves rhythm, breathing, discipline, relaxation, and so on, creates the graphical forms, which are 2-dimensional intersections of 3-dimensional paths in time, so 4 dimensions total. There is an amazing movie by a mathematician (Thomas Banchoff) showing a 4-dimensional cube, a tesseract, moving through 3-space, and of course projected into 2-space on a screen. At one screening years ago, the audience cheered as they understood what was happening. I think this is analogous to the mystical aspects of calligraphy — those two higher dimensions in 3- and 4-space, that we can infer from the 2-D graphical forms. Of course, there are good calligraphers and type designers who don't believe in the mystical aspects, who care only about the 2-D images, but the higher dimensions can enrich our appreciation.

For typography, the problem of revival is more like signal processing: how to distinguish signal from noise? The hand motions of the punch cutter are not important because the fonts are a kind of shallow sculpture, bas-relief. The engraved contours are the important things. For some typefaces, like some 16th century cuts by Garamond and Granjon, and 18th century cuts by John Handy, who cut type for Baskerville, we can get a very good idea of the signal because their hand-cut steel punches survive, and sometimes their matrices, the impressions that the punches make when driven into a blank of copper. But, reproducing the face of the punch is not a perfect solution for today, because the old punchcutters compensated a little bit for subsequent processes, especially ink-squash of type on the paper. The problem is much harder for typefaces for which no punches, matrices, or old type survives, like the types of Jenson, Aldus, and Fournier. There are two ways of doing things. First, let's preserve the original printed form as much as possible. This includes some noise along with the signal.

Y: Then you will get very ugly fonts.

C: Yes, I think so, but people who like this approach don't think it's ugly; rather, they say these imperfections preserve the feeling of ancient typography.

Y: Just like metal type typesetting was dead years ago, but now Apple makes letterpress cards for customers. Sometimes old-fashioned things get revived.

C: Exactly. We like the imperfections of the old methods because they have more personality than our modern methods, which often seem to lack soul, despite their advantages. The second approach is to re-interpret the type by trying to understand the intentions of the original type artist, and the limitations of the medium, and then reinterpret those intentions in modern media. A vision scientist who studies reading once told me that he doesn't really care how a typeface is made or printed — what he cares about is the image on the retina of the eye; that is what is communicated to the brain.

Y: Why not make the revival as authentic as the original one?

C: Sometime you can't. Between the old days of early type and digital type now, technology has changed, from wood block to metal type to mechanical type to phototype to digital type, from paper to CRT screens to LCD screens, to e-ink. And aesthetics and taste have also changed. In Europe, from old-style typefaces like Garamond, to modern styles like Bodoni, to sans-serifs like Helvetica. After high resolution digital typography made well-rendered classical designs cheap and easy, young designers in the 1990s rebelled against perfection and used “grunge” types and “distressed” types, full of dirt, errors, jaggies, and other noise.

In the KangXiZiDianTi you mentioned, the earliest printed editions were cut in wood blocks, I assume. Is that correct? First, some calligraphers had to write every character in the dictionary on paper, as models for the wood-cutter. Second, probably many wood-cutters cut the characters in wood blocks for printing. I don't know if individual characters were cut in wood, small individual pieces of wooden type, or a whole page on a single block. I think the latter, a whole block per page. After the calligraphers have died and their original handwritten examples transferred to wood and lost in history without a trace, there's no chance you can find them. Today, you cannot find the earliest wood blocks either. Third, after the wood cutting, there was the printing process — which slightly deforms the glyph shape as the ink is squeezed onto paper. And, after many impressions, the characters on wood become worn and less distinct. The same is true for metal type. The paper that early printers used, the ink they made, all have effects on the image in the final book. There may be more “signal” information lost in these processes. Fourth, the book preserved to this day may not look the same as it was hundreds of years ago. The humidity and temperature of the environment may change the glyph shape as well, not to mention disasters like insects eating the paper.

Fifth, during the photocopy and scanning process conducted today, there might also be other information losses. I believe that scanning at 600 pixels per inch is not enough to capture all the artistic information in text sizes of type. 1200 pixels per inch is much better, but it takes more time, and thus costs more. Even at 1200 ppi, the image is not perfect, because of noise. Then, if you scan with some level of gray depth, later you may have to threshold down to bi-level pixels for fitting curves around contours, although there are also methods of fitting contours to gray-scaled images.

Y: So it's not possible to trace down the original shape any more.

C: Exactly. In the western world we have already known that since the late 19th century, when English typographers began to revive old types by enlarging photographs of old books from the so-called "cradle" of printing, the years before 1501. Well before the digital era, it was recognized that data about the image was being lost in analog restorations. So most designers did a kind of creative renovation instead of trying to just remove noise from enlargements of the original form.

Y: I see.

C: Take the Jenson text as an example. I showed you a page typeset in Jenson earlier (fig. 3). Jenson was the first great creator of the "humanist" roman types that became the model for all subsequent printing in the Latin alphabet, but all his original punches, matrices, and types have been lost. A great pity. So all Jenson reproductions are redesigns from the images in books that Jenson printed from 1470 to 1480. There is a wide variation of weights and shapes of modern types modeled on Jenson.

Bruce Rogers' Centaur, created in 1914–15, is crisp and sharp, designed by reworking photographic enlargements with a pen. It was re-cut by Monotype in 1929 in a range of sizes. Morris Fuller Benton's Cloister of 1913 is darker and sturdier, made by engineering-style drawings based on enlargements, with attention to mechanical letterpress printing of the early 20th century. Robert Slimbach's Adobe Jenson of 1996 is a careful reconstruction for digital typography. Their variations show the visual senses, goals, and artistic or technical limitations facing modern designers.

Y: I know this typeface [Centaur], it was created for the Metropolitan Museum of Art.

C: Yes, that's true, so you know. Centaur has a lighter, sharper quality than Adobe Jenson, perhaps reflecting Rogers' pen-inscribed approach. It is rather light in digital imaging today because it was designed for letterpress printing, which added weight because of ink-squash. Ron Arnholm's Legacy of 1993 is Jenson modernized to late 20th century taste, with a larger x-height than the original. George Abrams' Venetian, circa 1999, is another careful and noteworthy Jenson revival. Also there is Hermann Zapf's Aurelia of 1983, for digital typesetting, which has some of the calligraphic accents of Palatino.

A: Body sizes the same

Centaur (Rogers 1915) [14 point]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&1234567890

Adobe Jenson (Slimbach 1996) [14 point]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&1234567890

Breughel (Frutiger 1981) [14 point]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&1234567890

B: x-heights the same

Centaur [17.45 point]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&1234567890

Adobe Jenson [16.45 point]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&1234567890

Breughel [14 point]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz&1234567890

Figure 30: Comparison of Jenson revivals.

I include among Jenson revivals Adrian Frutiger's most intriguing Breughel design of 1981. (See fig. 30 and also `http://odaddyo.com/typography/type_class/FrenchOldstyle.pdf`.) It is not concerned with imitating superficial aspects of Jenson's types, but is a deep attempt to render the philosophical spirit of Jenson's era, when handwriting was reduced to sculpture and mechanical reproduction. Breughel was released by Linotype 501 years after the death of Jenson. I think typographers weren't ready for such an innovative design in 1981. Maybe fifteen years later, in the era of punk and grunge typography, Breughel might have become more popular — unusual, a bit irregular, but legible. The Swiss typographer, Bruno Pfaffli, who was Frutiger's studio partner for many years, used it very well in catalogs and posters for French museums.

Y: No two of them look exactly identical.

C: And this is good in some sense — rather than restricting modern type designers to historical details, the new digital reproduction processes give them the freedom to create something new. So such a discussion in China, regarding the Kangxi dictionary characters, is wonderful. I am really happy to hear of such a discussion if it turns out to generate new thoughts and ideas.

Y: Thank you very much, Prof. Bigelow, for taking the time to do this interview. I have learned a number of things I didn't know. And many thanks also for your great contribution in digital type design and research, especially the work of the Lucida typeface family.

C: Because of its high legibility, I'm happy to see Lucida on computer platforms like Apple's Mac OS X, Sun's Java platform, Bell Labs' Plan 9 and Microsoft Windows. We're working on a web site for B&H at `http://www.lucidafonts.com` which will have (even) more information. For now, let's close by mentioning a very different example of Lucida in use — in a Colorado corn field in 2002 on the Fritzler farm; the corn plants are used as "pixels" to render Lucida Handwriting: `http://www.fritzlermaze.com/mazes.html`.

Acknowledgments

Many thanks to Prof. Charles Bigelow for doing this interview. Many parts of this talk have Prof. Kris Holmes' contributions as well. Without their help this interview would not be possible. Thanks to Xuan Zhang for introducing me to Prof. Bigelow. Jiang Liu provided the funding and invested a lot of resources to edit and publish this interview. A few questions in this interview were collected from Jiang Jiang, Rex Chen, Karl Berry, Luigi Scarso, and Shiang-Ing Ji.

References

[1] Huda Smitshuijzen AbiFares, *Arabic Typography*, Saagi Books, London, 2001.

[2] Charles Bigelow and Kris Holmes, "The design of a Unicode font", *Electronic Publishing*, **6**:3 (1993), 289–305. `http://cajun.cs.nott.ac.uk/compsci/epo/papers/volume6/issue3/bigelow.pdf`

[3] Charles Bigelow and Kris Holmes, "Notes on Apple 4 Fonts", *Electronic Publishing*, **4**:3 (1991), 171–181. `http://cajun.cs.nott.ac.uk/compsci/epo/papers/volume4/issue3/ep050cb.pdf`

[4] Hans Hagen, Taco Hoekwater, Volker RW Schaa, "Reshaping Euler: A collaboration with Hermann Zapf", *TUGboat*, **29**:2 (2008), 283–287. `http://tug.org/TUGboat/tb29-2/tb92hagen-euler.pdf`

[5] Yannis Haralambous, *Fonts & Encodings*, O'Reilly, Sebastopol, CA, 2007.

[6] Gordon E. Legge and Charles A. Bigelow, "Does print size matter for reading? A review of findings from vision science and typography", *J. Vis.*, **11**:5 (2011) 8. `http://www.journalofvision.org/cgi/content/abstract/11/5/8`

[7] Robert A. Morris, Kathy Aquilante, Charles Bigelow, and Dean Yager, "Serifs slow RSVP reading at very small sizes, but don't matter at larger sizes", Submission to Symposium session "Legibility and Usability Issues for Text Displays". SID Symposium Digest of Technical Papers, **33**:1, pp. 244–247, May 2002. `http://www.cs.umb.edu/~ram/rsvp/publications/SerifsSubmittedV2.doc`

[8] Vaughan Pratt, "Techniques for Conic Splines", ACM SIGGRAPH Computer Graphics, **19**:3, July 1985, pp. 151–160. `http://dl.acm.org/citation.cfm?id=325225`

[9] Emil Ruder, *Typographie*, Verlag Arthur Niggli, Sulgen, Switzerland, 1967. `http://www.designers-books.com/typography-emil-ruder-1967/`

◇ Yue Wang
`yuleopen (at) gmail dot com`