

**Additional notes
for
Font Wars, Parts 1 and 2
by Charles Bigelow
IEEE Annals of the History of Computing
Volume 42, Number 1, 2020**

Copyright © 2020 Charles Bigelow

Additional notes for part 1: [Note 1](#) ; [Note 2](#) ; [Note 3](#) ; [Note 4](#) ; [Note 5](#) ; [Note 6](#) ; [Note 7](#) ; [Note 8](#) ; [Note 9](#) ; [Note 10](#) ; [Note 11](#) ; [Note 12](#) ; [Note 13](#) ; [Note 14](#) ; [Note 15](#) ; [Note 16](#) ; [Note 17](#) ; [Note 18](#) ; [Note 19](#) ; [Note 20](#) ; [Note 21](#) ; [Note 22](#)

Additional notes for part 2: [Note 23](#) ; [Note 24](#) ; [Note 25](#) ; [Note 26](#) ; [Note 27](#) ; [Note 28](#) ; [Note 29](#) ; [Note 30](#) ; [Note 31](#) ; [Note 32](#)

PDF format of this document
(may display better on some devices)

Part 1 additional notes

[Note 1]

The traditional meaning of “character” in typography has both symbolic and graphical senses because typographic characters convey two sorts of information. One is the symbols of a writing system. The other is the graphical forms representing the symbols. The symbols are abstract, significant elements of a language or other symbolic system. The graphical shapes representing the symbols have concrete, visible forms. Modern digital fonts map character encodings to graphical forms. Tapping a key on a computer keyboard starts a chain of mappings that eventually result in a visible form on a screen or in print.

This distinction is explicit in the Unicode character encoding standard, which assigns numerical computer codes to the abstract characters or symbols of writing systems. Unicode does not encode or specify the forms of those characters, which it calls “glyphs” — specific shapes.

Applying the classic Shannon-Weaver model of communication, digitization encodes the source forms of typographic characters for transmission, and reception decodes them for the destination, the eyes — or more comprehensively the human visual and cognitive systems in reading. In the Font Wars, most of the source forms were analog typefaces.

Digital encoding, transmission, and decoding introduces noise, such as under-sampling or “aliasing” when encoding forms, or distortions when decoding in display and printing, such as blurring. Experienced readers have viewed hundreds of millions of letters through decades of education and reading, and hence are skilled at recognizing a wide variety of character forms and renderings under different reading conditions, but nevertheless expect certain familiar featural characteristics. Some typographers distinguish between a definition of “legibility” that means

letter forms can be recognized and text can be read, versus “readability,” which means that forms of text are not only recognizable but also pleasurable to read. These terms are not used consistently, however, whether by typographers or by reading researchers, so their aggregate sets of meanings intersect and often seem synonymous. Noise in the typographic image may impair pleasure in reading, even if measurable functionality like speed of reading is not significantly diminished.

Thus, in the Font Wars, technical questions were coupled with vision questions. Technically: How to encode analog letter forms into digital font data? How to compress font data to conserve computer memory? How to devise efficient algorithms to increase speed of decoding? Visually: What spatial resolutions are adequate to represent familiar analog letter forms? How can digital text be faithful to traditional type aesthetics? How can it satisfy readers’ expectations of legibility? Of readability? Pleasurable, or merely functional?

A few of these issues are noted by Robert Sproull in describing fonts for the pioneering Alto display and laser printing at Xerox PARC in the 1970s:

“The Xerox Alto Publishing Platform,” *IEEE Annals of the History of Computing*, vol. 40, no. 3, July-September 2018, pp. 38–54.

As digital typography began to compete with analog typography, an implicit assumption was: if the forms of digital letters fall below traditional standards of typographic legibility, then the basic transmission of literate information fails. In most cases, however, substandard digital type did not functionally fail; readers could read it but didn’t like it, and migrated to higher resolutions and higher quality. The failure was in the market.

[Note 2]

Ideas can have long histories. Cuneiform tablets expressing the most ancient of the ideas involved in the Font Wars were written anonymously on clay tablets around four thousand years ago and were transmitted through neighboring cultures and successive civilizations, Sumerian, Babylonian, Elamite, and others. Many of those ancient mathematical texts were presumably lost when their respective civilizations collapsed but some were rediscovered thousands of years later through archaeology in the 19th and 20th centuries. Some of the ideas, however, may not have been entirely lost in ancient times, but were transmitted to Greece, where philosophical and mathematical discoveries were culturally valued and identified by the names of their discoverers or transmitters, such as the Pythagorean theorem, Thales’ theorem, and the Elements of Euclid. The idea of naming discoverers may have been an extension of the ancient Greek concept of “kleos” or fame.” We know the “imperishable fame” of Achilles in the Iliad, where it is called “kléos áphthiton” = “κλέος ἄφθιτον”. The name of the Muse of History, Clio or Kleio, “(she) who makes famous, glorifies,” comes from the same root as kléos, meaning “fame,” “glory”.

In addition to honoring war heroes, the Greeks gave enduring fame to their mathematicians and scientists. We know the names and discoveries of Thales, Pythagoras, Euclid, and Archimedes not by accident, but by culture. Ancient Babylonian mathematicians discovered the Pythagorean relationships of triangles and areas but are anonymous in surviving texts. Though the Pythagorean “relationship” was discovered than 1,500 years before Pythagoras, it is Pythagoras whose name and fame have endured because of the Greek tradition of naming and faming. [See

Note 7.]

P. S. Rudman, *The Babylonian Theorem: The Mathematical Journey to Pythagoras and Euclid*, Prometheus Books, 2010.

An echo of the Greek culture of fame recurs in the Italian Renaissance with its culture of Humanism, classics, and focus on mankind, when discoverers of new ideas (or old ideas long lost) were rewarded with fame. We know of Fibonacci, Pacioli, Tartaglia, and Cardano, among mathematicians. In modern times, the linking of discovery to discoverers is communicated through scientific journals, the first of which was probably the *Philosophical Transactions of the Royal Society*, founded in 1665. Through publications in scientific journals, we can see how the ideas underlying Bézier curves were communicated by successive mathematicians and computer scientists for a century until the curves showed up in fonts. Thus, in recounting a history of the Font Wars, we can follow the Muse of History and confer some fame on the discoverers and contenders.

[Note 3]

The pages of digitized type and ornaments of Lamartine's *Les Laboureurs* were woven on a Jacquard loom by the lace-making firm of J.-A. Henry in Lyon, France. One of the very rare woven copies of *Les Laboureurs* is in the Cary Graphic Arts Collection of the Rochester Institute of Technology. It is dated 1883. A copy dated 1878 is in the Musée des Tissus in Lyon, France. Another copy is in the US Library of Congress, and another is in the Bibliothèque Nationale de France.

The Jacquard punched cards were apparently stored for several years and reused from time to time, hence the different dates of issue. We do not know how the book was “programmed,” and the cards are presumably lost. In the Cary Collection issue of *Les Laboureurs*, there a slight bitmap differences between different instances of the same letters on a page, which suggests that the pages were probably digitized on the basis of printed texts rather than composed from a “font” of pre-digitized letters. The name(s) of whoever digitized/programmed the pages — a laborious task — appear to be unknown, and likewise those who did the actual weaving are also unknown.

Many more details have been published about a similarly digital but later and elaborately illustrated and ornamented book woven by the same method by the same firm, J.A. Henry; this is the *Livre de Prières tissé d'après des Enluminures des Manuscrits du XIVe au XVI Siècle*. Designed by P. Hervier and published in 1886 by A. Roux, it was hailed as an “artistic marvel” and won a prize at the 1889 Universal Exposition in Paris, where the newly completed Eiffel Tower was a celebrated phenomenon.

The *Livre de Prières* comprises fifty pages of digitized Gothic text and ornamentation based on late medieval illuminated manuscripts. Its digital resolution is roughly 400 pixels per inch. The number of Jacquard cards has been estimated as between 100,000 to 500,000 Jacquard cards.

“Une Merveille Artistique: Un Livre de Prières Tissé en soie.” *La Correspondant*, 1889, pp. 602–604.

An informative, recent essay on the history of the *Livre de Prières*, comparing several issues of it, is:

Matthew Westerby, *The Woven Prayer Book: Cocoon to Codex*, Satellite Series, Paris & Chicago: Les Enluminures, 2019.

An earlier history of the *Livre de Prières* and its manuscript models, with particular attention to the issue at the Walters Art Gallery, is:

Lilian M.C. Randall, "A Nineteenth Century 'Medieval' Prayerbook Woven in Lyon," in *Art the Ape of Nature*, M. Barasch, L. F. Sandler, P. Egan, eds. New York: Harry N. Abrams, pp. 651-668.

These rare, woven digital books were marvelous tours-de-force but were so labor intensive in "programming" and production that they were not competitive with standard analog printing.

Digital text was also used for titling in an 1839 digital portrait of Joseph-Marie Jacquard, who had developed punched cards to control weaving with a mechanical loom. The portrait was digitally encoded by Michel-Marie Carquillat using 24,000 cards, and was woven by the Didier, Petit et Cie. firm in Lyons. A physical example of this rare digital portrait is in the New York Metropolitan Museum of Art.

<https://www.metmuseum.org/toah/works-of-art/31.124/> (Accessed June 10, 2019)

Charles Babbage, who intended his prospective Analytical Engine to be programmed with Jacquard punched cards, admired the high-resolution bitmap portrait of Jacquard and obtained a copy.

Referring to Babbage's Analytical Engine, Ada Augusta, Countess of Lovelace, often said to have been the first programmer because of her lucid writings on Babbage's invention, wrote: "We may say most aptly that the Analytical Engine *weaves algebraical patterns* just as the Jacquard loom weaves flowers and leaves."

Had Babbage's Analytical Engine been constructed and used to render digital images and text, it is faintly conceivable that outline fonts defined by polynomial equations could have been rasterized and woven as bitmaps, a century before the Apple LaserWriter used polynomial outline fonts.

J. Essinger, *Jacquard's Web: How a hand loom led to the birth of the information age*, Oxford U. Press, 2004.

[Note 4]

Warde's reference to the Andromeda galaxy plays on the title of Marshall McLuhan's *The Gutenberg Galaxy*, which analyzes the effects of typography on culture and society:

M. McLuhan, *The Gutenberg Galaxy: The Making of Typographic Man*, University of Toronto Press, 1962.

Warde is best known for her essay, “The Crystal Goblet or Printing Should be Invisible.”

B. Warde, *The Crystal Goblet: Sixteen Essays on Typography*, H. Jacob, ed., London: Sylvan Press, 1955.

A few modern, or modernist, critics have said that Warde’s “The Crystal Goblet” is too conservative, too focused on readability in book typography. In “A Twinkle In Andromeda,” however, Warde was light-years ahead of her critics. Astronomers tell us the Andromeda galaxy is due to collide with our Milky Way around 4.5 billion years from now, an event that may profoundly affect our Gutenberg Galaxy. But, Beatrice doesn’t have to wait until then; she has already beamed aboard.

[Note 5]

In summer 1965, Dr.-Ing. Rudolf Hell announced the development of a raster based CRT typesetter. A prototype machine was shown in February 1966 at a Hannover printing trade show in Germany, and a production version, the Digiset 50T1 digital typesetter, was installed at a telephone directory printer in Copenhagen near the end of 1966 (a few sources state 1967). In 1967, RCA Graphic Systems Division arranged to market the Hell Digiset hardware (“back-end”) with RCA software and computer controller (“front-end”), as the VideoComp system. By the 1970s, Digiset fonts were based on a raster grid of 120 horizontal units and 100 vertical units. Around 1971, RCA sold the VideoComp system to Information International, Incorporated (variously abbreviated “Triple-I” or “III”) which eventually developed its own hardware and software.

A CRT digital typesetter developed for IBM by the Alphanumeric firm, the APS-3 (Alphanumeric Photocomposition System), was sold by IBM for use with System 360 and became operational in 1969. Alphanumeric and the APS line were eventually sold to the Autologic corporation, which developed several later APS machines, notably the APS-5, which in several versions was widely used for newspaper, magazine, and directory typesetting.

To reduce the burden of producing and storing digital fonts for different sizes, some digital typesetting machines used lenses to enlarge or reduce digital output within a certain range, and some could increase or decrease the spacing of the vertical scanning CRT beam, thus scaling the letters in the horizontal dimension and proportionally scaling the vertical distances of the beam runs. Using these methods, early Digiset machines used two master fonts to set type from 4 to 24 point, and later four to five master fonts. Later Videocomp machines used five master fonts to set type from 4 to 96 point. Autologic APS machines usually used four digital font masters for a full range of sizes.

J. W. Seybold, *The World of Digital Typesetting*, Seybold Publications, 1984.
computerhistory.org/collections/catalog/102740425

Those scaling work-arounds were not feasible in laser printers, for which bitmap fonts were produced for each type size. A method of off-line scaling of bitmap fonts to generate different

sizes for laser printers was devised by Casey, Friedman, and Wong and used to make fonts for some IBM laser printers, but those fonts were generated off-line and usually required some hand-editing after generation. The method does not appear to have been used on-the-fly in output devices.

R. G. Casey, T. D. Friedman, and K. Y. Wong, "Automatic scaling of digital print fonts," *IBM Journal of Research and Development*, vol. 26, no. 6, 1982, pp. 657-666.

In 1976, the Monotype Corporation launched the Lasercomp System 3000 with raster fonts and a raster-image processor (RIP) that could output full pages of digital type and images. In 1983, Adobe Systems experimentally drove a Lasercomp with PostScript before concentrating on development of the LaserWriter printer for Apple.

[Note 6]

Bitmap screen displays.

Early raster graphics screen displays were developed in research laboratories, in particular Bell Telephone Laboratories and Xerox PARC. The first appears to have been by A. Michael Noll in 1971 at Bell Telephone Laboratories.

A.M. Noll, "Scanned-display computer graphics," *Communications of the ACM*, vol. 14, no. 3, 1971, pp. 143-150.

In 1972, Tom Knight at the MIT AI Laboratory designed a semiconductor bitmap display, later commercialized in a newspaper layout system.

In the mid-1970s, bitmap raster displays were developed at Xerox PARC.

R. F. Sproull, "The Xerox Alto Publishing Platform," *IEEE Annals of the History of Computing*, vol. 40, no. 3, July-September 2018, pp. 38-54.

The Xerox 8010 Star Information System launched in 1981 was the first commercial product to incorporate bitmap raster display coupled with bitmap laser printing.

J. Seybold, "Xerox's 'Star'," *The Seybold Report, Media*, PA: Seybold Publications, vol. 10, no. 16, 1981.

"Designing the Xerox "Star" User Interface," *Byte*, issue 4, 1982, pp. 242-282.

By 1983, bitmap displays were common on workstations and terminals. A raster display terminal adapted to Unix systems was developed at Bell Labs in 1982-1983.

R. Pike, "The UNIX system: The blit: A multiplexed graphics terminal," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 8, 1984, pp. 1607-1631.

Slow reading on screens.

John D. Gould and others at IBM found that reading speeds were slower on screens than on

paper, but the reasons were not clear. In further studies, involving proofreading, Gould et al. found that as certain characteristics of screen text on CRT displays more closely resembled text on paper, the differences in reading speed diminished. Readers still preferred paper, however.

J. D. Gould, L. Alfaro, V. Barnes, R. Finn, N. Grischkowsky, and A. Minuto, "Reading is slower from CRT displays than from paper: attempts to isolate a single-variable explanation," *Human Factors*, 1987, 29(3), pp. 269-299.

J. D. Gould, L. Alfaro, R. Finn, B. Haupt, and A. Minuto, "Reading from CRT displays can be as fast as reading from paper," *Human Factors*, 1987, 29(5), pp. 497-517.

A review by Dillon examined potential causes of slower reading on screens. Possible causes included: reading distance; text line lengths and visual angle; screen flicker; text polarity; character size; font choice, among other factors. Following Gould, Dillon suggested that a combinations of factors rather than a single factor may account for screen versus print differences. Noting the disparity between different displays, Dillon suggested that the differences between reading on screen and paper were likely to continue until there were higher standards of text display quality.

A. Dillon, "Reading from paper versus screens: A critical review of the empirical literature," *Ergonomics*, 35(10), 1992, 1297-1326.

Gray-scaling.

A popular method of ameliorating the jaggies and irregularity of letter forms in low-resolution displays is to render characters in levels of black, gray, and white, instead of only bilevel black or white. This has been called gray-scaling, anti-aliasing, multi-level rendering, and smoothing. All assume a display system capable of rendering a range of pixel intensities.

A straight-forward approach, in conception if not always in implementation, is to calculate the area of an edge pixel that lies inside the contour of a character (the parts that are black in traditional print), and to adjust the pixel's gray level proportionally. If an edge pixel is wholly inside a contour, it is black; wholly outside, white; if half inside and half outside, then middle gray. The more area inside the contour, the darker the tone; the less area inside, the lighter the tone. There are many ways of achieving this.

A detailed description and illustration is in a patent by J. Valdés & E. Martinez. "Raster shape synthesis by direct multi-level filling," United States Patent 5,438,656, August 1, 1995.

Other gray-scaling approaches include filtering and convolution.

A. C. Naiman, "The use of grayscale for improved character presentation," Ph.D. Thesis, University of Toronto, 1991.

Gray-scaling of pre-grid-fitted characters and legibility testing is described by Morris, Hersch, and Coimbra.

R. A. Morris, R. D. Hersch, and A. Coimbra, "Legibility of condensed perceptually-tuned grayscale fonts," In *International Conference on Raster Imaging and Digital Typography*, Berlin, Heidelberg: Springer, 1998, pp. 281-293.

Readers have often judged grayscaled text to look better than jagged bilevel black/white bitmaps at the same display resolution. Several studies, however, have not found appreciable advantages in reading performance. Some have been reviewed by Gordon Legge.

G. E. Legge, *Psychophysics of Reading in Normal and Low Vision*. Mahwah, NJ: Lawrence Erlbaum Associates, 2007, pp. 123-125, 132.

"Sub-pixel rendering" is a form of anti-aliasing on LCD (Liquid Crystal Display) screens, but cannot be used on CRT screens, so it did not become common until after the Font Wars. Instead of adjusting the gray-value intensity of a full pixel, as on a CRT screen, sub-pixel rendering separately adjusts the intensities of the red, green, and blue sub-pixels that constitute a full pixel, thus enabling fine-tuning of character edges and features. Microsoft's Advanced Reading Technologies group developed a version of sub-pixel rendering into a technology called ClearType. In the early 2000s, ClearType showed some promise in reading aesthetics and possible performance improvements, as reviewed by Kevin Larson.

K. Larson, "The technology of text," *IEEE Spectrum*, 2007, 44(5), 26-31.

The findings of some other studies have been mixed. Gugerty et al. found some improvement of reading performance:

L. Gugerty, R. A. Tyrrell, T. R. Aten, & K. A. Edmonds, "The effects of subpixel addressing on users' performance and preferences during reading-related tasks," *ACM Transactions on Applied Perception*, 1(2), 2004, pp. 81-101

But, Sheedy et al. found that ClearType did not improve reading performance, in terms of legibility, reading speed or comfort, although subjects preferred moderate ClearType rendering to other forms of grayscale.

J. Sheedy, Y. C. Tai, M. V. Subbaram, S. Gowrisankaran, & J. R. Hayes, "ClearType sub-pixel text rendering: Preference, legibility and reading performance," *Displays*, 29(2), 2008, 138-151.

In the 21st century display technology shifted from CRT to LCD (Liquid Crystal Display) screens with multi-level, color pixels, and display resolutions that rose from around 100 pixels per inch on CRTs to above 200 pixels per inch on the LCD screens on laptop computers and above 300 ppi on smart phones (above 400 ppi on some phones). On these high resolution screens, when type is rendered with gray-scaling algorithms, jagged edge pixels and character irregularities are usually imperceptible.

[Note 7]

Mesopotamian mathematics.

The eminent historian of ancient mathematics, Otto Neugebauer discussed Old Babylonian cuneiform mathematical texts using triangles and polygons inscribed in circles, presumably to

determine radiuses and circumferences. Neugebauer also discusses traces of Babylonian mathematics that appear in Greek treatises some 1,500 years later. O. Neugebauer, *The Exact Sciences in Antiquity*, Brown University Press, 1957. Particularly chapter II, section 23-25 and notes.

Some Mesopotamian mathematical texts discussed by Neugebauer were found in Susa, not Babylon, and first described by E. M. Bruins, "Quelques Textes Mathématiques de la Mission de Suse," Koninklijke Nederlandsche Akademie van Wetenschappen, 1950, pp. 1025-33.

In more recent decades, there has been a resurgence of studies of Mesopotamian mathematics, including the long-standing question of how much did Babylonian (or generally, Mesopotamian) mathematics influence Greek mathematics. Answers are varied, ranging from not much, to somewhat, to a lot.

Peter Rudman proposes the methods that Old Babylonian mathematicians may have used to calculate the value of pi by using polygons. He also demonstrates a method that Archimedes used to calculate pi more than a thousand years after Babylonian and Egyptian mathematics.

Peter Rudman, *The Babylonian Theorem: The Mathematical Journey to Pythagoras and Euclid*, Prometheus Books, 2010.

Jens Høyrup gives a detailed comparison of the Mesopotamian understanding of the Pythagorean "rule" to the Greek Pythagorean "theorem."

Høyrup, J. (1999). Pythagorean "rule" and "theorem": mirror of the relation between Babylonian and Greek Mathematics. In J. Renger (Ed.), *Babylon : Focus mesopotamischer Geschichte, Wiege früher Gelehrsamkeit, Mythos in der Moderne: 2. Internationales Colloquium der Deutschen Orient-Gesellschaft 24.-26. März 1998 in Berlin* Saarbrücken: SDV Saarländische Druckerei und Verlag GmbH.

Elsewhere, Høyrup argues that a direct connection between Mesopotamian mathematics and Greek mathematics is doubtful.

J. Høyrup "Mesopotamian Mathematics," in *The Cambridge History of Science*, Vol. 1: Ancient Science, A. Jones and L. Taub, eds., Cambridge: Cambridge University Press.

Egyptian mathematics.

Also in *The Cambridge History of Science*, Vol. 1, in "Egyptian Mathematics," Høyrup discusses the discrepancy between claims of Greek historians, ranging from Herodotus to Proclus, that early Greek mathematicians, particularly Thales and Pythagoras, learned geometry in Egypt, compared to modern research in Mesopotamian mathematics that suggests that the Greeks may have learned mathematics from Babylonian sources.

Høyrup offers a possible reconciling of the conflicting histories, by suggesting that the Assyrian conquest of Egypt (in 7th century BCE) may have transmitted Mesopotamian mathematical knowledge to Egypt, where a century or more later, early Greeks learned it and assumed it

originated in Egypt.

Some Egyptian mathematics long predates the Assyrian conquest of Egypt, however. The Rhind Papyrus, a mathematical treatise copied by the scribe Ahmes (Ahmose) around 1550 BCE, is a tutorial and compilation of mathematical problems. To calculate the volume of a cylindrical granary, it gives a close approximation to π (π) using the fraction of $256/81$ (3.1605), and similarly gives a close approximation of the area of a circle compared to the area of a circumscribed square.

The Greek philosopher Proclus, writing in the 5th century CE, a thousand years after the time of Thales, claims that Thales studied geometry in Egypt and brought the knowledge back to Greece. A further link, according to Proclus, is that Thales encouraged Pythagoras to study in Egypt. Proclus states that Pythagoras was a student of Anaximander, who was a student of Thales, thus the connection between Thales, Pythagoras, and Egypt. It is believed that Proclus was familiar with a history of geometry written by Eudemus of Rhodes, who was a student of Aristotle in the 4th century BCE. Eudemus' history has since been lost.

In favor of the Mesopotamian-Greek transfer of mathematical knowledge, Joran Friberg examines correspondences between Mesopotamian and Greek mathematics in *Amazing Traces Of A Babylonian Origin In Greek Mathematics*, Singapore: World Scientific, 2007.

A history of Mesopotamian mathematics and their relationships to ancient societies and cultures, is *Mathematics in Ancient Iraq*, by Eleanor Robson, Princeton: Princeton University Press, 2008.

Whether any of the ancient claims of connections of Babylonian and Egyptian mathematics to Greek mathematics are true remains a matter of conjecture, archaeology, research, and debate today, since most of the ancient written sources are lost, if they ever existed, and there are gaps of centuries between fragmentary accounts. Nevertheless, it is evident that Greek historians believed in the transmission of ideas between cultures and furthermore provided plausible linkages between known persons and shared knowledge, demonstrating ancient “nonrivalry” of ideas, in Paul Romer’s model.

Moreover, the ancient Greek concept that the path of transmission of knowledge can be reconstructed as a string of academic or literate connections between named philosophers is not dissimilar to documented transmissions of mathematical knowledge in modern times, although our documentation is better (in part because it is much more recent, and ignoring disputes over priority) and our cycle time is much faster. Take, for instance, Sergei Bernstein’s approximation theory paper on which Bézier curves are based. Bernstein’s doctoral thesis advisers were Émile Picard and David Hilbert, leading mathematicians of the era. Bernstein’s paper that led to Bernstein-Bézier curves in computer graphics was a proof of a theorem by 19th century mathematician Weierstrass. There is a gap between Bernstein’s paper of 1912 and Paul de Casteljau’s application of it to computer graphics. How he found Bernstein’s paper is unclear. In the 1970s, A. R. Forrest recognized the equivalence of Bézier’s methods with Bernstein’s mathematics. And so on, until Bézier curves were used for fonts. The transmission of knowledge was much faster and better documented by typography in modern times than by cuneiform tablets and papyrus manuscripts in the ancient world, but the structure of transmission is similar.

Greek mathematics and letter forms.

Swetz (1996) recounts a claim by a 2nd century CE Greek grammarian, Apollonius of Messene (not Apollonius of Perga), that Pythagoras “sought to achieve visual harmony within each letter through a systemic use of angles, line segments and circular arcs.” This assertion was made more than 600 years after the presumed death of Pythagoras, and it seems more like myth than history, but at least it indicates that the idea of geometric letter construction occurred in the late Hellenistic era.

From the 5th century BCE onward, most Greek inscriptions of any length were cut by professional stone cutters who apparently worked fast and freehand with chisel and maul, carving letters on the fly without the aid of geometric construction. Stephen V. Tracy has noted, however, a rare instance of a late 6th century BCE Athenian inscription in which the circular letters Theta and Omicron were inscribed with some kind of compass or drill.

S. V. Tracy, *Athenian Lettering of the Fifth Century BCE*, Berlin/Boston: De Gruyter, 2016, pp. 17-18.

B. F. Cook, *Greek Inscriptions*, Berkeley: University of California Press, 1987.

In a later and presumably coincidental instance of transfer of Mesopotamian literate forms to Greek culture, Stanley Morison, typographic scholar and co-designer of Times New Roman, observed that the wedge shaped terminals of letters in certain Ionian Greek inscriptions imitated the wedge shaped elements of cuneiform inscriptions of the older Persian and Assyrian cultures. Morison argued that imitation of cuneiform graphical features in Greek alphabetic letters was intended to confer the imperial prestige of older civilizations on Hellenistic Greek civilization. He also asserted that the wedge terminals were the ancestors of modern serifs.

S. Morison, *Politics and Script: Aspects of Authority and Freedom in the Development of Graeco-Latin Script from the Sixth Century BC*. Oxford: Oxford University Press, 1972.

[Note 8]

The “Hershey fonts” were intended for preparation of mathematical reports and were freely distributed in the early era of computer graphics. Hershey’s technical report was distributed to Bell Labs, the American Mathematical Society, the Mergenthaler Linotype company, National Geographic Society, university geographers, NASA, CIA, National Bureau of Standards, US Government Printing Office, among other military and governmental agencies.

A. V. Hershey, “Calligraphy for computers,” U.S. Naval Weapons Laboratory, Dahlgren, VA, Report No. NWL-TR-2101, 1 August 1967.

In 1968, Mergler and Vargo produced a proof-of-concept parametric demonstration of polygonal outlines of twenty-four capital letters that could be transformed by adjusting parameters. The results could be output on a digital plotter, but were not expanded into a practical font.

H. W. Mergler and P. M. Vargo, “One approach to computer assisted letter design,” *Visible*

Language, 2(4), 1968, pp. 299-322.

[Note 9]

The Linotron 202, introduced in 1978, was the first major typesetter to use polygonal outline fonts and was one of Linotype's most successful machines: 13,000 were sold. Its fixed horizontal output resolution was 972 scan lines per inch and its vertical resolution - the number of addressable start/stop points per vertical scan - was 486 pixels per inch (although other numbers were also **stated**). At text sizes, the polygon sides of characters were close enough to the 202's output resolution that ink-spread on newsprint smoothed letter edges.

Two Linotype patents on the 202 character method show a letter 'Q' digitally mapped into a grid 60 x 80 units, but it is not clear if that is the actual stored resolution of the stored characters, or simply a convenient example.

Linotype patented the 202's polygonal character generation method twice.

U.S. Patent No. 4,199,815 of April 22, 1980, "Typesetter Character Generating Apparatus." The inventors were Derek J. Kyte, Walter I. Hansen, and Roderick I. Craig.

U.S. Patent No. 4,254,468 of March 3, 1981, "Typesetter Character Generating Apparatus." The inventor was Roderick I. Craig.

Derek Kyte was involved in the design and development of most of Linotype's digital typesetters from the Linotron 505 to the 202. In 1985, he joined other former Linotype engineers a new firm, Chelgraph, which made raster image processors.

(Linotype's name at the time was "Mergenthaler Linotype" and its corporate parent was a conglomerate, Eltra Corporation, **to which the patent were assigned**. Linotype later went through several name changes and corporate configurations, but for brevity and convenience, it is referred to simply as "Linotype" here.)

The Linotype patents cites reviews stating that a font of polygonal outline letters was used in the Seaco 1601 digital typesetter (reviewed in the Seybold Report in Vol. 1, Nos. 12 and 13 (Feb. 14 and 28, 1972 respectively). The Seaco company apparently went bankrupt later in 1972, so it is not clear if the machine was ever commercially operational.

In the summer of 1979, a Linotron 202 was purchased by the Computing Science Research Center at Bell Labs, where its software was reverse-engineered and reinvented by Lab scientists Jos Condon, Brian Kernighan, and Ken Thompson, who were dissatisfied with the limitations of Linotype's original 202 software. They reinvented and extended the Linotron 202 software because they wanted the machine to work the way they thought it should, for efficient typesetting of technical reports, patent applications, books, and other documents and publications of Bell Labs.

Condon, Kernighan, and Thompson devised and installed a new operating system for the 202, decoded the 202's secret polygonal font format, created and uploaded their own characters and

fonts including the Bell symbol, chess characters, a monospaced font with a full ASCII character set, mathematical characters, and rotated font for labeling axes on graphs. They wrote programs for the machine to draw lines, arcs, figures and charts for technical publishing, and wrote other software tools for the machine. They built a hardware-software interface between the 202 and Bell Labs' DEC PDP/11 minicomputer (running Unix, of course) to bypass Linotype's cumbersome paper tape input. They summarized their work in a 1980 technical report:

J. Condon, B. Kernighan, K. Thompson, "Experience with the Mergenthaler Linotron 202 Phototypesetter, or, How We Spent Our Summer Vacation," Bell Laboratories, Computing Science Technical Memorandum. January 6, 1980.

Joe Condon was a physicist who made significant discoveries in solid-state electronics and was more widely known as the co-inventor with Thompson of "Belle," a chess playing computer that was the North American computer chess champion in several years and the world computer chess champion in 1980. Brian Kernighan co-wrote with Dennis Ritchie *The C Programming Language* (an edition of which is still in print after 40 years), wrote the device-independent Unix typesetting software *ditroff* (initially to drive the 202), wrote many other software tools, co-wrote other programming languages, and authored or co-authored several books on programming, and software, and Unix, including *Unix: A History and a Memoir*. Ken Thompson is the co-inventor of Unix with Dennis Ritchie, for which they jointly received the Turing Award of the Association for Computing Machinery (ACM), the Hamming Medal of the Institute of Electrical and Electronics Engineers (IEEE), and other awards. Thompson is also a co-inventor of the programming language Go and operating system Plan 9, with Rob Pike and others.

When Linotype learned of the Labs' innovative software for the 202, they feared that disclosure of the secret font encoding format could enable widespread digital piracy of proprietary Linotype fonts. It was a real concern. Linotype fonts had been widely pirated in the phototype era, so Linotype was wary of a repeat in digital type. Around 90% of Linotype's revenues came from typesetting machine sales and only 10% from font sales, but Linotype's highly respected fonts were the reason that most Linotype customers bought the machines, so fonts were crucial to Linotype's business. Therefore Linotype asked Bell Labs to not distribute the research software for the 202 and to suppress publication about the work, fearing it might lead to decodings by others of the font format.

Accordingly, the "Summer Vacation" report was not published, although knowledge of it was passed on by oral tradition. Three decades later, the original memo was reconstructed and published in PDF format by Steve Bagley and David Brailsford at the University of Nottingham, working with Brian Kernighan.

S. R. Bagley, D. F. Brailsford, and B. Kernighan, "Revisiting a summer vacation: digital restoration and typesetter forensics," *Proceedings of DocEng 2013*, Florence, Italy, 2013.

A scanned copy of the original report and the reconstructed report by Bagley, Brailsford, and Kernighan, is online at:

<http://www.cs.princeton.edu/~bwk/202> (accessed June 6, 2019)

An online narration of the story by Brailsford is at:
<https://youtu.be/CVxeuwlvf8w> (accessed June 6, 2019)

Linotype's concerns about digital font piracy were well founded. Extensive digital font piracy did eventually occur, but a dozen years later and involving Adobe's PostScript Type 1 fonts, not Linotron 202 fonts. Moreover, piracy of Adobe fonts affected fonts potentially used on millions of computers and printers, not the 13,000 Linotron 202s. [See Note 31.]

[Note 10]

Renaissance geometrical construction of Roman capitals combined the Humanistic revival of Roman culture with renewed interest in Euclidean geometry. Ancient Roman inscriptions survived in Renaissance Italy, and Euclid's *Elements* in manuscript was a standard subject in the university curriculum called the Quadrivium (its four parts were the arts of number: arithmetic, geometry, music, and astronomy). Hence, Renaissance scholars would have been familiar with Euclidean construction by compass and straight-edge, as well as with Roman capital inscriptions.

Nevertheless, Edward Catich's demonstration of the primacy of brush-written roman capitals shows a more plausible method than geometric construction:

E. M. Catich, *The Origin of the Serif: Brush Writing and Roman Letters*, St. Ambrose, IA: Catfish Press, 1968.

One reason that geometric construction was not used for printing types is that they were too small. Italian Renaissance roman printing types had body sizes around 12 to 18 points in modern measurement. The master patterns of type - the "punches" - were cut by hand at actual font sizes by skilled artisans — "punchcutters." It would have been impractical to first construct, then reduce the geometric letter shapes to the face of a punch, and then cut the form exactly, a punch being one letter of one style of one size of type. Another reason that construction was not used for type is that the illustrated geometric constructions were of capital letters, whereas Renaissance typographic books were largely composed in lowercase, following the models of Humanist manuscripts. There were few large models for the shape of lowercase. Dürer made constructions of a narrow, formal "textura" style of blackletter, but most Renaissance Italian and French books, and later English books, were in the type style we call "roman" today.

[Note 11]

Moxon, Joseph, 1627–1691: *Regulæ Trium Ordinum Literarum Typographicarum: or The Rules of the Three Order of Print Letters: viz. The {Roman, Italick, English} Capitals and Small. Shewing how they are compounded of Geometrick Figures and mostly made by Rule and Compass. Useful for Writing Masters, Painters, Carvers, Masons, and others that are Lovers of Curiosity.* By Joseph Moxon, Hydrographer to the Kings most Excellent Majesty. London: Printed for Joseph Moxon, on Ludgate Hill at the Sign of Atlas. 1676)

A digitized on-line version is at:
<https://archive.org/details/regulaetriumordi00moxo/page/n6>

Moxon's better known book is *Mechanick Exercises: Or, the Doctrine of Handy-Works Applied to the Art of Printing*, which is the earliest known manual of printing and is still studied today by print historians. Reprints include:

Moxon, Joseph. *Moxon's Mechanick Exercises or the Doctrine of Handy-Works Applied to the Art of Printing: A Literal Reprint in Two Volumes of the First Edition Published in the Year 1683*, With Preface and Notes by Theo. L. De Vinne. New York: The Typothetae of the City of New-York, 1896.

Joseph Moxon, *Mechanick Exercises on the Whole Art of Printing*, H. Davis and H. Carter, eds. Oxford University Press, 1958. 2nd edition, 1962. There are several other reprints as well as ebooks of the work.

[Note 12]

The Royal Committee was led by Abbot Jean-Paul Bignon and included Gilles Filleau des Billettes, royal printer Jacques Jaugeon, and mathematician Jean Truchet, known as Fr. Sébastien Truchet. They used microscopes to study the proportions and shapes of existing typefaces, including some cut by Robert Granjon a century earlier. The Committee did not copy the shapes of earlier types, however. Instead, they created new letter forms constructed by compass and ruler at large size on gridded backgrounds, ranging from 384 x 384 units, up to 487 x 486 units for ascending or descending lowercase letters. The model letters were engraved on copper plates by Louis Simonneau beginning in 1695, and in later years.

The size of the grid on which the initial constructions of the Romain du Roi were engraved was approximately 7 centimeters square, but the size of the eventual printing types were to be around 5 to 6 millimeters in body height, a reduction to 8 percent of the original. There was no automatic way to scale the large designs to the small printing type size. Instead, the “punches” — the steel master patterns — were hand-cut at final type sizes by Philippe Grandjean, an artisan of whom relatively little is known beyond his work on the Romain du Roi.

Thus, the idea of geometrically defining letter outlines as curves and lines in a Cartesian coordinate system was demonstrated around 280 years before polynomial curved outlines were devised for digital type contours. The problem for the savants of the Bignon committee was that in 1695, and for centuries thereafter, was that there was no automatic way to scale large master designs down to type sizes for the printing technology of the day. It had to be done by hand and eye.

From the first decades of printing in the 15th century until the end of the 19th century, printing types were first hand-cut as steel “punches” which were then hammered into copper “matrices” from which type was cast. The King's Roman was no exception. It has been argued that in hand-cutting the punches, Grandjean reinterpreted the Committee's idealized geometric letter models into practical type forms, but a counter-argument is that Grandjean's punches are remarkably faithful to the large models. In any case, the influentially elegant fonts first used in 1702 in an elegant book honoring the reign of Louis XIV, *Médailles du règne de Louis XIV*, 1702.

Another innovation of the Romain du Roi was that its italic letter models were produced by slanting the upright grids by approximately 14 degrees, predating by almost three centuries the

automatic “oblique” types generated in digital typography. Furthermore, the italic letter forms utilized tight circular arcs for serifs and stroke terminals, reinterpreting the concept of “cursive” forms.

Although use of the Romain du Roi types was restricted to the French Crown, their innovative style inspired French and Italian designers, particularly Pierre-Simon Fournier, Giambattista Bodoni, and the Didot family, later in the 18th century, culminating in a style today called “Modern”.

Until Linn Boyd Benton’s invention of the pantographic punchcutting machine in 1885, there was no mechanical technique for accurately reducing large type drawings to small metal sizes, although pantographic cutting machines were used for large wooden type. For the Romain du Roi, Philippe Grandjean cut punches for the French-named sizes of Cicero (12 point), St. Augustin (14 point), Gros Romain (16 point) and Petit Parangon (18 point); fonts of these sizes were cast for the 1702 printing of a book celebrating the “Medals of Principal Events in the Reign of Louis XIV.” After Grandjean, more sizes of the Romain du Roi family were cut by Jean Alexandre and Louis Luce. All are now in the archives of Imprimerie Nationale of France. A digital revival of Grandjean’s hand cut types of the Romain du Roi (not copies of the engraved drawings) was made by Franck Jalleau in 1995–1997, for exclusive use of the Imprimerie Nationale.

References:

A. Jammes, *La Réforme de la Typographie Royale sous Louis XIV: Le Grandjean*. Paris: Librairie Paul Jammes, 1961.

A. Jammes, *La Naissance d'un caractère : le Grandjean*, Paris: Cercle De La Librairie, 1989.

J. André, D. Girou, “Father Truchet, the typographic point, the Romain du roi, and tilings,” *TUGboat*, vol. 20, no. 1, 1999, pp. 8–14.

online at: <https://tug.org/TUGboat/tb20-1/tb62andr.pdf>

[Note 13]

Around 1982, Rockwell sued Eltra Corporation, the conglomerate that owned Linotype, claiming that the polygonal font outline method of the Linotron 202 typesetter infringed Rockwell’s outline font patent. Eltra settled out of court for around 6 million dollars. Rockwell sold the patent to Information International, Inc., which then sued Compugraphic Corporation claiming infringement by the Compugraphic 8600 typesetter. Compugraphic reportedly settled for 5 million dollars. In May, 1989, Information International sued Adobe and Apple, claiming infringement by Adobe’s PostScript Bézier cubic curve outline font technology in PostScript. After four years of litigation and millions of dollars in expenses, Adobe prevailed in 1993, first in trial court and then in Federal District Appeals court, obviating future claims based on the Rockwell patent. A victory by Triple-I might have altered the course of the Font Wars and slowed innovation in digital font technology.

The Evans and Caswell patent is:

G. W. Evans and R. L. Caswell, “Character generating method and system,” U.S. Patent 4,029,947, 14 June 1977; Reissued: RE30,679, 14 July 1981.

William Schreiber's review of the background of the Triple-I patent is:

W. Schreiber, "Outline Coding of Typographical Characters," Printing Industries of America, 1987. [Abstract at: <https://www.printing.org/taga-abstracts/outline-coding-of-typographical-characters> (accessed June 6, 2019)]

[Note 14]

An example of an idea that was "in the air" yet with some historical record, is the use of cubic curves for input and storage of character outlines, with conversion to arc and vector formats for output and distribution. This approach was developed first by Peter Karow in his Ikarus system at URW in late 1972 or early 1973, using Hermite form cubic curves, based on an algorithm from Helmuth Späth:

Späth, H: "Spline-algorithmen zur Konstruktion glatter Kurven und Flächen," Oldenburg Verlag München Wien, 1973.

A progression from cubic splines to arc and vector format occurred over time in the research of Philippe Coueignoux and his students. Coueignoux's 1973 MIT M.S. thesis defined character outlines with cubic spline approximations from De Boor and Rice:

C. De Boor and J. R. Rice, "Least squares cubic spline approximation, II - variable knots," Computer Science Dept., Purdue University, Tech. Report CSD TR 21, 1968.

In 1978, Coueignoux's student Marc Hourdequin used circular arcs in his Ph.D. thesis:

M. Hourdequin, "Génération de polices d'Imprimerie pour photocomposeuse digitale," Dr.-Ing. Thesis, École Nationale Supérieure des Mines de Saint-Étienne, Institut National Polytechnique de Grenoble, 1978.

In 1981, Coueignoux's student Marc Bloch used circular arcs in his Ph.D. thesis:

M. Bloch, "Génération de taches bicolores : application aux caractères d'imprimerie; problèmes de nature ordinale," Thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 1981.

A related approach was used in the Camex-Bitstream Letter IP developed in 1981–1982:

J. Flowers, "Digital Type Manufacture: An Interactive Approach," IEEE Computer, May 1984.

Camex principals knew of Philippe Coueignoux's MIT research [24], had met with him, and learned of the circular arc doctoral research by his students in France (Hourdequin and Bloch cited above). Bitstream's founders had previously worked at Linotype in New York, which had installed an Ikarus system.

The Intellifont format developed by Thomas Hawkins at Compugraphic also used a circular arc and vector format derived from the Ikarus format, familiar to Compugraphic as a licenser of the Ikarus system.

United States Patent 4,675,830, T. B. Hawkins, inventor. June 23, 1987

[Note 15]

Written around 225 BCE, the texts of the first four of Apollonius of Perga's books were preserved from the Hellenistic era through the Middle Ages into the Renaissance. Translated from Greek to Latin by Giovanni Memo, they were printed in Venice by Bernardino Bindoni in 1537. The next three books, 5–7, were lost in the original Greek but survived in medieval Arabic translations and were translated into Latin by Giovanni Borelli and Abraham Ecchellensis and were printed by Cocchini in 1661. In the 11th century, Persian astronomer and mathematician Omar Khayyam (of Rubaiyat fame) studied Euclid and Apollonius of Perga in Arabic and used intersections of conics to solve cubic equations. The text of Apollonius' eighth and final book was apparently lost entirely, but Edmund Halley (of comet name and fame) speculatively reconstructed it from fragments and references, and published it in Latin in 1710. In 1896, Thomas Heath translated and edited Apollonius' work on conics into an English edition, still available in reprints today. A 2012 English translation of Halley's reconstruction of Apollonius' eighth book is by Michael Fried. Although Apollonius appears not to have been read widely, the stature of mathematicians and scholars who have admired and interpreted his works suggest that he was rightly called "the great geometer" in his day, as Heath has said.

As with polygons and circles, conic curves were not used in ancient, classical, or Islamic lettering. Hand-methods were faster and cheaper and met the needs of the literate cultures. Even in the typography of Halley's day, calculation of curves was irrelevant to the cutting of type forms. Even if the Bignon savants, particularly mathematician Truchet, had made the effort to calculate conic curve outlines for fonts, there would have been no way to implement them in printing types. That would not be achieved until the research by Pratt in 1985, and the commercial implementation by Folio around 1990, almost 300 years later.

References to Apollonius and Khayyam:

E. Halley, *Apollonii Pergaei Conicorum Libri Octo et Sereni Antissensis de Sectione Cylindri & Coni, Libri Duo*: Oxoniae, e Theatro Sheldoniano, 1710. (Dual language texts: Latin and Greek: Book Eight of the Conics by Apollonius of Perga; and Two Books on the Section of the Cylinder and Section of the Cone, by Serenus of Antissa. Oxford, 1710.)

T.L. Heath, *Apollonius of Perga: Treatise on Conic Sections*. Cambridge: Cambridge University Press, 1896.

M. N. Fried, *Edmond Halley's Reconstruction of the Lost Book of Apollonius's Conics: Translation and Commentary*, New York: Springer, 2012.

Omar Khayyam, *An Essay By The Uniquely Wise 'Abel Fath Omar Bin Al-Khayyam on Algebra and Equations* (Algebra wa Al-Muqabala), R. Khalil, trans. London: Garnet, 2009.

[Note 16]

The "splines" of thin wood or metal strip used in aircraft lofting produce curves of least energy, that is, the least internal strain, according to Berthold Horn:

B.K.P. Horn, “The Curve of Least Energy,” ACM. Transactions on Mathematical Software, Vol. 9, No. 4, December 1983, pp. 441–460.

Horn notes a suggestion that the human visual system may use a virtual curve of least energy when constructing a subjective contour. The trend in computer graphics, however, went to B-splines and parametric cubic curves.

I.J. Schoenberg, “Contributions to the problem of approximation of equidistant data by analytic functions, Part A. — On the problem of smoothing or graduation. A first class of analytic approximation formulae.” *Quarterly Applied Mathematics* IV, 1946, pp. 45–99.

Following Schoenberg’s research, “Basis splines” or “B-splines” were applied to computer graphics research in academic, government, aircraft, and automotive research. A. R. Forrest [12] discusses splines as part of a broad survey and development of mathematical methods in computer-aided design. The first use of B-splines in fonts appears to be that of Coueignoux [20], who described letter outlines using cubic splines from De Boor and Rice.

Further references:

C. De Boor and J. R. Rice, “Least Squares Cubic Spline Approximation II,” Technical Report CSD TR 21, Computer Science Dept., Purdue University, April 1968.

Gordon & Riesenfeld discuss B-splines in interactive graphics.

W. J. Gordon and R. F. Riesenfeld, “B-spline curves and surfaces,” in *Computer Aided Geometric Design*, Academic Press, 1974, pp. 95-126.

Quadratic B-splines define parabolic arcs and are equivalent to quadratic Bézier curves, as Casselman (2008) points out, referring to the curves used in TrueType as quadratic Bézier curves.

The conic sections can be algebraically described by quadratic equations, in which the unknown variable is squared, hence the term “quadratic” from Latin “quadrus” meaning a square, or four.

A review of Schoenberg’s 1983 book, *Mathematical Time Exposures*, can be found online at:

http://www-history.mcs.st-and.ac.uk/Extras/Schoenberg_time_exposures.html
(accessed September 30, 2019)

[Note 17]

The first international conference on computer aided geometric design was held at the University of Utah on March 18 to 21, 1974. The proceedings of this influential conference were published as *Computer Aided Geometric Design*, edited by Robert Barnhill and Richard Riesenfeld. Authors included Pierre Bézier, A. R. Forrest, Martin Newell, as well as the two editors and others whose research influenced the future development of computer graphics.

R. Barnhill and R. Riesenfeld, Eds., *Computer Aided Geometric Design*. New York: Academic Press, 1974.

A graphical irony of this volume of leading-edge computer graphics is that its typography used strike-on technology, not digital computer typesetting. Nearly all the articles were set by typewriter, predominantly the IBM Selectric, and the primary font was Prestige Elite, designed at IBM's typewriter division in 1953 by Clayton Smith (a few sources say Howard Kettler, the designer of the perennially favorite font Courier).

Prestige Elite is a modern, monospaced interpretation of a seriffed typeface cut by Francesco Griffo in 1495 for a book published by renowned Renaissance scholar-printer, Aldus Manutius. Aldus, a scholar of Latin and Greek, was so proud of his new font that he wrote a short Latin poem praising “grammatoglyptae” (“carved letters”) cut by “the Daedalus-like hands of Francesco of Bologna.” Aldus appears to have coined the Latinized Greek term *grammatoglyptae* (carved letters), which was descriptively accurate but not generally adopted. The English term “fount” or “font” (from Latin “fundere” — to pour, cast, through French “fonte” into English) appeared in print in Joseph Moxon's treatise on printing in 1683. The term “type,” although in earlier use with various meanings, does not seem to have been used for printing type until the early 18th century.

The Aldus-Griffo type was admired in the French Renaissance, and in the early 1530s, a few French punchcutters, most famously, Claude Garamond, cut refined versions of the Griffo-Aldine face, which influenced the look of typefaces for centuries to come. The likely direct inspiration for Prestige Elite is the typeface Bembo, a 1929 revival of the Griffo-Aldine roman by the Monotype Corporation. Prestige Elite is a long way from the Aldine roman, much transformed by adaptation to the technology of the typewriter, but some details of Griffo's style can still be detected in it. Alas, digital versions of Prestige Elite (and its larger size, Prestige Pica) ignore the subtle features traceable back to Bembo or Griffo. For a book of leading-edge mathematics and computer science to be printed with a font designed two decades earlier for typewriters, and nearly 500 years earlier in the early decades of printing, continues the longstanding division that, while mathematicians of many cultures — Babylonian, Greek, Renaissance, and modern — ponder the higher forms of thought, their words are reproduced by humble craftsmen — scribes, punchcutters, and typewriter font designers.

After the Utah conference on computer aided geometric design in 1974, the Association for Computing Machinery (ACM) held its first SIGGRAPH (Special Interest Group on Computer Graphic), conference in Boulder, Colorado. Titles of the varied range of presentations are on the SIGGRAPH web site: <https://dl.acm.org/citation.cfm?id=563182&picked=prox> (Viewed June, 2019)

[Note 18]

Bézier curves - brief chronology of ideas and implementations.

- 1885 - Karl Weierstrass paper: proof of a theorem in approximation theory
- 1912 - Sergei Bernstein paper: constructive proof of Weierstrass' theorem
- 1959 - Paul de Casteljau - cubic Bernstein curves in computer-aided automobile design, and

recursive drawing algorithm, at Citroën.

- 1960 - Pierre Bézier - cubic curves in computer aided automobile design, at Renault [later recognized by Forrest as equivalent to curves defined by Bernstein polynomials]
- 1966–1967 - Pierre Bézier papers: Définition numérique des courbes et surfaces I & II
- 1969 - A. R. Forrest paper: Re-examination of the Renault technique for curves and surfaces
- 1972 - A. R. Forrest paper: Interactive interpolation and approximation by Bézier polynomials
- 1974 - Pierre Bézier paper: Mathematical and practical possibilities of UNISURF
- Bézier curves known to Xerox PARC researchers by mid-1970s
- 1982 - Warnock & Wyatt paper: SIGGRAPH - Bézier curves in imaging model
- 1984 - Knuth software: Metafont 84
- 1985 - Adobe book: PostScript Language Reference Manual
- 1990 - Adobe book: Adobe Type 1 Font Format

Pierre Étienne Bézier was an engineer at the French Renault automobile company, where from 1960 to the mid-1970s, he developed methods of computer modeling of curves and surfaces for automobile design. Paul de Faget de Casteljau was a mathematician at rival French automaker Citroën, where beginning in 1959, he developed a method of modeling curves and surfaces with cubic parametric curves based on the work of Sergei Bernstein. At this remove, it is unclear how De Casteljau came to use cubic curves based on an approximation theorem of Bernstein.

Sergei Natanovich Bernstein, born in Ukraine, studied mathematics at the Sorbonne in Paris, and for a year at Göttingen with David Hilbert, before returning to Paris to write his 1904 doctoral dissertation, with Émile Picard, Jacques Hadamard, and Henri Poincaré on his thesis committee. (By coincidence, Picard had married the daughter of mathematician Charles Hermite, whose interpolatory method was much later adopted by Karow for Ikarus format fonts.)

Around 1912, Bernstein wrote the paper that became the basis of Bézier curves in computer graphics. Published in French in the *Communication of the Kharkov Mathematical Society*, Bernstein's paper proved an 1885 theorem by Karl Weierstrass on approximation theory. Bernstein's proof was simple and constructive, providing a practical method of computation that may have contributed to its future appeal to Paul De Casteljau in computer graphics.

Around 1959, De Casteljau devised an elegant recursive sub-division algorithm to digitally calculate and draw Bernstein curves, but Citroën did not permit De Casteljau to publish, though some of his work was circulated within Citroën. Renault did permit Bézier to publish descriptions of his research. Hence, Bernstein-based parametric curves were named for Bézier. In the 1970s, Wolfgang Böhm tracked down the prior work of De Casteljau, and his recursive sub-division algorithm was named De Casteljau's algorithm.

A. R. Forrest discusses the mathematics of Bézier curves in a technical report and following papers (1969, 1972, 1974). The 1969 technical report was prompted by a demonstration at General Motors of the method Bézier had developed at Renault. Forrest periodically visited and lectured at the computer graphics group at University of Utah, as did Bézier, notably in 1974. Forrest also visited the Xerox PARC computer science lab and was a visiting scientist there in 1982–1983. Through Forrest's publications and lectures, and those of Bézier and others, the mathematics of Bézier curves became well known in computer graphics.

[Note 19]

In the 1982 SIGGRAPH paper by Warnock and Wyatt, general shapes and characters of fonts were described with Bézier curves. That model was based on previous work by Warnock and Martin Newell, as well as Wyatt, at PARC, and on prior work at Evans and Sutherland. After the founding of Adobe, Warnock implemented a similar imaging model using Bézier curves in the PostScript graphics language. PostScript was designed to handle all the kinds of shapes encountered in graphic arts, not just fonts, so the generality of Bézier curves, including curvature-continuity, was a factor in choosing a single outline description for every shape including characters of fonts.

General advantages claimed for Bézier curves are that: they require a comparatively small number of on-curve and off-curve control points, thus providing better data compression than quadratic B-splines, conics, or circles; they are easily renderable into polygons and pixels on raster devices using De Casteljau's algorithm; they are numerically stable in computation; they provide continuity of curvature when pieced together to approximate more complex curves, thus enabling smooth-looking shapes, in particular the contours of letter forms.

Commercial success of Adobe's PostScript graphics language and use of Bézier curves in Adobe Illustrator led also to use of Bézier curves in font editing and design software. Because Bézier curves can be manipulated interactively with relative ease on computer screens and provide continuity of curvature as well as tangency when curve segments are joined, they are favored for interactive font design.

A 1984 photo of John Warnock with whiteboard drawings of Bézier curves includes a diagram of De Casteljau's algorithm for evaluating Bernstein cubic polynomial curves. A portion of the photo was featured on the cover of Adobe's PostScript Language Reference Manual in 1985.

Bézier curve fonts are easily converted to the quadratic curve format of TrueType, usually called quadratic B-splines but sometimes referred to as quadratic Bézier curves.

References:

Historical

S. N. Bernstein [or Bernshteĭn], "Démonstration du Théorème de Weierstrass fondée sur le calcul des Probabilités," *Comm. Soc. Math. Kharkov* 2. Series XIII No.1, 1912.

K. Weierstrass, "Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen," *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften* zu Berlin, 1885, II, pp. 633–639 & 789–805.

Modern

Pierre Bézier, "Procédé de définition numérique des courbes et surfaces non mathématique - système UNISURF automatisme," vol. 13, no. 5, May 1968.

G. Farin, "History of Curves and Surfaces in CAGD," *Handbook of CAGD*, G. Farin, MS Kim, J.

Hoschek, editors.

R. T. Farouki, “The Bernstein polynomial basis: A centennial retrospective,” *Computer Aided Geometric Design*, vol. 29, no. 6, 2012, pp. 379–419.

A. R. Forrest, “A Re-examination of the Renault technique for curves and surfaces,” Computer-Aided Design Group, University of Cambridge, CAD Group Document No. 24, 1969.

A. R. Forrest, “Interactive interpolation and approximation by Bézier polynomials,” *The Computer Journal*, vol. 15, no. 1, 1972, pp. 71–79.

A. R. Forrest, “Computational geometry — achievements and problems,” in *Computer Aided Geometric Design*, R.E. Barnhill and R.F. Riesenfeld, Eds. New York: Academic Press.

C. Rabut, “On Pierre Bézier's life and motivations,” *Computer-aided design*, 34(7), 2002, 493–510.

J. Warnock and D. K. Wyatt, “A device independent graphics imaging model for use with raster devices,” *ACM SIGGRAPH Computer Graphics*, vol. 16, no. 3, 1982.

J. Warnock, “Simple Ideas That Changed Printing and Publishing,” *Proceedings of the American Philosophical Society*, vol. 156, no. 4, 2012, pp. 363–378.

J. Warnock, “The origins of PostScript,” *IEEE Annals of the History of Computing*, vol. 40, no. 3, 2018, pp. 68–76.

Corporate

Adobe Systems Inc. *PostScript Language: Reference Manual*. Mountain View, CA: Adobe Press, 1985. (Later editions: Addison-Wesley)

Adobe Systems Inc. *Adobe Type 1 Font Format*. Addison-Wesley Longman, 1990.

[Note 20]

In *The METAFONTbook* (1986, p. 13) Knuth wrote: “The recursive midpoint rule for curve-drawing was discovered in 1959 by Paul de Casteljau, who showed that the curve could be described algebraically....This polynomial of degree 3 ... is called a Bernshtein polynomial because Sergeĭ N. Bernshtein introduced such functions in 1912 as part of his pioneering work on approximation theory. Curves traced out by Bernshtein polynomials of degree 3 are often called *Bézier cubics* after Pierre Bézier who realized their importance for computer aided design during the 1960s.”

Knuth described Metafont as it evolved over several years.

D. E. Knuth, “Metafont: A System for Alphabet Design,” Stanford Computer Science Department, Report No. STAN-CS-79-762, 1979.

D. E. Knuth, “The concept of a meta-font,” *Visible Language*, vol. 16, no. 1, 1982, pp. 3-27.

D. E. Knuth, *Computers and Typesetting, Volume C: The Metafontbook*. Addison-Wesley, 1986.

D. E. Knuth, *Computers and Typesetting, Volume D: Metafont: The Program*. Addison-Wesley, 1986. [The code is online at: <https://www.ctan.org/tex-archive/systems/knuth/dist/mf/> (accessed June 10, 2019)]

D. E. Knuth, *Computers & Typesetting, Volume E: Computer Modern Typefaces*. Addison-Wesley, 1986.

Knuth's "Metafont for Lunch Bunch" was attended by Stanford students, faculty, and visiting scholars, who influenced digital typography in both Latin and non-Latin typography.

Of relevance and interest concerning the mathematics related to Metafont are the thesis and papers by Knuth's doctoral student, John Hobby.

J. D. Hobby, "Digitized brush trajectories," Ph.D. thesis, Department of Computer Science, Stanford University, 1985. [<https://tug.org/docs/hobby/hobby-thesis.pdf>]

J. D. Hobby, "The MetaPost User's Manual," [<https://tug.org/docs/metapost/mpman.pdf>]

Applications of Metafont to Latin and non-Latin typefaces include:

J. Hobby and G. Gu, "A Chinese meta-font," Stanford Computer Science Department, Report No. STAN-CS-83-974, 1983.

P. K. Ghosh and D. E. Knuth, "An approach to type design and text composition in Indian scripts," Stanford Computer Science Department, Report No. STAN-(X83-965), 1983.

D. Wujastyk, "The many faces of TeX: A survey of digital METAfonts," TUGboat vol. 9, no. 2, 1988, pp. 131–151.
<https://tug.org/TUGboat/tb09-2/tb21wujastyk.pdf>

D. Wujastyk, "Further Faces," TUGboat 9(3), 1988, pp. 246–251.
<https://tug.org/TUGboat/tb09-3/tb22wujastyk.pdf>

A. M. Sherif and H. A. Fahmy, "Meta-designing parameterized Arabic fonts for AlQalam," TUGboat vol. 29, no. 3, 2008 pp. 435–443.
<https://tug.org/TUGboat/tb29-3/tb93sherif.pdf>

[Note 21]

In a 1965 essay, Gordon Moore observed that the number of transistors in an integrated circuit doubled around every two years.

G. E. Moore, "Cramming more components onto integrated circuits," *Electronics* 38(8) 1965.

A decade later, Moore revised his estimate.

G. E. Moore, “Progress in Digital Integrated Electronics” from Technical Digest 1975. International Electron Devices Meeting, 21, 1975, pp. 11–13.

Although not a physical law, Moore’s observation was dubbed “Moore’s Law” and was confirmed over several decades of increase in computer chip densities and decrease in chip prices.

The Apple LaserWriter, in which fonts were Bézier curve outlines, was introduced in 1985 at a price around \$7,000. It included three families of four styles of fonts plus a symbol font, all capable of being output at virtually any size. In price comparison, in 1973, an RCA VideoComp digital typesetter with a resolution of 600 dots per inch, would have cost around \$875,000 converted to 1985 dollars. Additional costs for a line drawing module, page rotation capability, more memory, a console terminal, a film developing unit, and fonts, would have boosted the price to around one million dollars, in 1985 dollars.

[Note 22]

In automobiles, continuous surface curvature seems to be desired more for cosmetic than aerodynamic reasons. Discontinuities of curvature can be visually detected on reflective automobile surfaces, but continuity of curvature of exterior surfaces is hardly functional for aerodynamics. In average daily commutes in Silicon Valley, aerodynamically-smooth cars like Teslas, Ferraris, and Lamborghinis chug along clogged roads at speeds below 35 mph, where laminar airflow is not needed. The smooth surfaces of luxury autos appear to increase their appeal as ostentatious luxuries known as Veblen goods, for which demand increases as prices rise. (The term is based on Thorstein Veblen’s theory of conspicuous consumption). Smooth-looking curves may be beautiful but not necessarily functional.

This point was dramatized in the movie, “Ford v Ferrari,” when race car designer Carroll Shelby and driver Ken Miles, working for Ford, first see rival Ferrari’s smoothly sleek cars in the 1966 Le Mans 24 hour race.

Ken Miles : If this were a beauty pageant, we just lost.
Carroll Shelby : Looks aren't everything.
(Ford won the race.)

Letter forms are two-dimensional shapes that don’t need to be aerodynamic or, indeed, to move at all. In traditional reading, letters sit still while our eyes make very fast jumps called “saccades” to fixate on letter groups for fractions of a second. In RSVP (Rapid Serial Visual Presentation) speed reading technology, individual words of text are flashed onto a computer screen for very short times, so the reader’s eyes do not have to move, and neither do the words.

Typeface aesthetics have a 500 year history, and legibility research a 100 year history, yet it remains unclear if continuity of curvature in type forms is an issue in aesthetics or legibility. Some type connoisseurs, but not all, prefer fair curves, but the difference between fair and smooth curves is not defined in typography, although Forrest [12] offers a clear and useful distinction.

Detectable curvature discontinuity was rare in traditional analog fonts, unless intended by the punchcutter or designer, but visible discontinuity can be observed in constructed letters. Patrick Baudelaire, in discussing the Fred font editor he developed at Xerox PARC, stated that, “A common example of this effect [lack of continuity of curvature] is the case of a straight line segment (which has no curvature) connecting tangentially to a circular arc (which has constant curvature): the sudden jump of curvature, from zero to some fixed value, may be viewed, in certain applications, as aesthetically undesirable.”

P. Baudelaire, “The Xerox Alto Font Design System,” Visible Language vol. 50, no. 2, 2016.

There are historical and anecdotal reasons to doubt that continuity of curvature is necessary in type forms. In the photo-typesetting era, photo masks of letters for film negative fonts were numerically cut with Ikarus circular arc outline data, yet there seem to have been no complaints about lack of smoothness, although the analog photographic optics and imaging may have smoothed out apparent discontinuities of curvature. In TrueType fonts, curved contours are composed of piecewise parabolic arcs and lack continuity of curvature of joins, yet among thousands of different TrueType fonts, and billions of downloads of web fonts, there appear to be few complaints about discontinuity of curvature. Possibly, rasterized output on screens and printers filters out visual discontinuity of curvatures, and, moreover, the small sizes at which types are composed for continuous reading, usually less than 1/6 inch, may render such discontinuities below the acuity threshold of human perception.

Nevertheless, continuity of curvature of cubic splines dominates interactive design of fonts on computer screens. Most interactive font design and editing tools now use Bézier cubic curves as the primary curve-drawing technique, although such tools usually enable editing and output of quadratic curves as well.

A possible drawback of Bézier curves is that they can easily produce figures with self-interesting curves, twists, warbles, oscillations and other peculiarities not found in traditional fonts and useless or harmful — artistically or technically — in digital fonts. Type designers drawing with Bézier curves are therefore economical in using on-curve points (“knots”), conservative in positioning them, and careful in manipulating their off-curve control points (“handles”), in order to avoid visually or technically undesirable or dysfunctional contours. Hence, designers discard much of the versatility of Bézier curves, although continuity of curvature is preserved. A few type critics have complained that, as a result, too many Bézier-constructed typefaces look too smooth. Perceptibility of curve continuity, whether too much or too little, has not been investigated in vision science, so aesthetic opinion and mathematical taste have not been independently and scientifically reconciled.

[End of Part 1 notes]

Part 2 additional notes

[Note 23]

The alphabet was invented in the ancient Near East before the middle of the second millennium

BCE. By around 1000 BCE, it was used in the writing of Phoenician, Hebrew and Aramaic. Around 750 BCE (some scholars propose an earlier date), the Greeks borrowed the Phoenician alphabet and repurposed some of its consonant letters to represent Greek vowels.

The symmetries of our capital letters can be traced back to ancient Greek inscriptions. In the sixth century BCE, Athenian inscriptions in “boustrophedon” style were read right-to-left and left-to-right in alternate lines. Several letters then had enantiomorphic forms, that is, left-handed or right-handed, depending on the reading direction. In the fifth century BCE, “stoichedon” inscriptions were read strictly left to right, and several formerly enantiomorphic letter forms were redesigned with bilateral symmetry. In stoichedon, letters were equally spaced and arranged in columns as well as lines, as in modern monospaced fonts. By the 4th century BCE, two-thirds of the 24 standard Greek letters were bilaterally symmetrical; six others were either reflectively symmetric through the midline or rotationally though not bilaterally symmetric. The inscriptional letters were cut with consistent thickness and weighting of strokes, similar to modern sans-serif designs. The overall effect was a high degree of symmetry. The Etruscans borrowed the Greek alphabet around 600 BCE, and the Romans borrowed the Etruscan alphabet around 500 BCE. Some symmetry was lost with each borrowing, and by the time of Augustus Caesar, the Roman inscriptional alphabet had 23 letters, of which 10 were bilaterally symmetrical; eight others were either reflectively symmetric through the midline or rotationally but not bilaterally symmetric.

Edward Catich demonstrated that Roman inscriptional lettering, in particular the inscription carved on the base of the Trajan Column in Rome in 113 CE, was first brush-written with combinations of basic strokes and then chiseled into stone.

E. M. Catich, *The Origin of the Serif: Brush Writing and Roman Letters*, St. Ambrose, IA: Catfish Press, 1968.

In late Medieval and Renaissance Latin handwriting, the precursor to typography, the shapes of letters were inked traces of moving tools. A scribe writing with a pen, brush, or stylus imparted regularity, repetition, rhythm, phase, constraint, alignment, and symmetry to the pattern of text.

Our lowercase roman letters lost most of the bilateral symmetry of the classical capitals but innovated new rotational and reflective symmetries, as in b d p q and translational symmetry of strokes as in v w x. Our lowercase was derived from a style developed by scribes working in the court of Charlemagne before 800 CE, and revived by Renaissance humanist scribes around 1400 CE.

When the kinetic strokes of handwriting were supplanted by the sculpted forms of type, letters became abstract, carved shapes, not compositions of strokes. The master forms of letters were carved in steel by punchcutters and regulated by justifiers, who adjusted letter spacings and alignments of matrices before the letters were cast. Despite the radically different medium, typographic artisans inherited some symmetries and imposed on printing types regularities that have endured for five hundred years.

In the history of typography, several punchcutters explored abstract symmetry by cutting type ornaments, called “fleurons” or “printers’ flowers.” The flowers cut by Robert Granjon in the

16th century displayed particular ingenuity, inventiveness, intricacy of combination, and delicacy of punchcutting. He was called “Maestro Roberto” by Bodoni and “intagliatore di caratteri singularissimo” (most exceptional engraver of characters) when introduced to Pope Gregory XIII. Granjon’s flowers have been used and admired for 450 years in revivals in metal, photo, and digital fonts.

H. D. L. Vervliet, *Granjon’s Flowers*, New Castle DE: Oak Knoll Press, 2016.

H. D. L. Vervliet, *Robert Granjon, letter-cutter; 1513–1590*, New Castle DE: Oak Knoll Press, 2018.

In western mathematics, Johannes Kepler briefly noted plane symmetries in his 1619 treatise on harmonies in astronomy. In the 20th century, notable symmetry studies have been published by several mathematicians, though it must be admitted that they did not treat alphabetic forms nor have an appreciable effect on typography. Nevertheless, they illustrate intriguingly beautiful symmetries unexplored in typography.

J. Kepler, *Harmonices Mundi*, 1619.

A. Speiser, *Die Theorie Der Gruppen Von Endlicher Ordnung*, Berlin: Julius Springer, 1923.

H. Weyl, *Symmetry*, Princeton: Princeton University Press, 1952.

A. V. Shubnikov & A. V. Koptsik, *Symmetry in Science and Art*, tr. G. D. Archard, New York: Plenum Press, 1974.

T. W. Wieting, *The Mathematical Theory of Chromatic Plane Ornaments*, New York: Marcel Dekker, 1982.

B. Grünbaum & G. C. Shephard, *Tilings and Patterns*, New York: W. H. Freeman, 1987.

Recent studies of symmetry in culture include:

D. K. Washburn and D. W. Crowe, *Symmetries of Culture: Theory and Practice of Plane Pattern Analysis*, Seattle: University of Washington Press, 1987.

D. K. Washburn and D. W. Crowe, eds., *Symmetry Comes of Age: The Role of Pattern in Culture*, Seattle: University of Washington Press, 2004.

The enduring fascination with symmetry in typography is found in the border patterns created by Donald Knuth’s students in his course on Metafont.

D. E. Knuth, “A Course on METAFONT Programming,” TUGboat 5(2), 1984, pp. 105–118.
<https://tug.org/TUGboat/tb05-2/tb10knut.pdf>

Giambattista Bodoni, one of the most prolific and renowned punchcutters in history, placed “regularity” first among the qualities that make typefaces beautiful, and observed that typeface designs can be composed of a small number of parts:

“...the four different qualities from which, I think, all their beauty seems to come. The first is regularity. Analyzing the alphabet of any language, one not only can find similar lines in many

different letters, but will also find that all of them can be formed with a small number of identical parts, combined and disposed in various ways.”

G. Bodoni, *Manuale Tipografico*. Presso la Vedova, 1818, pp. XXI–XXVIII. (Facsimile reprint (2010), S. Füssel, ed. Köln: Taschen. This reprint includes supplementary English translation by H.V. Marrot: *G.B. Bodoni's Preface to the Manuale Tipografico of 1818: now first translated into English*, London: Elkin Mathews Ltd, 1925.)

A discussion of aesthetics and levels of typographic structure is:

C. Bigelow, "Form, Pattern, and Texture in the Typographic Image," *Fine Print*, 15(1), April 1989. <https://bigelowandholmes.typepad.com/bigelow-holmes/2015/04/form-pattern-texture-in-the-typographic-image.html>

Frank Blokland combined historical and metric research on surviving Renaissance types with aesthetic theory to elucidate the process of adaptation, standardization, and harmonization of letter forms, proportions, and spacing in the development of Renaissance types, which remain the models for many modern typefaces.

F.E. Blokland, "On the origin of patterning in movable Latin type : Renaissance standardisation, systematisation, and unitisation of textura and roman type," Ph.D. thesis, Leiden University, 2016.

[Note 24]

As an example of irregularity, the left stem of an 'n' could look thinner or thicker than the right stem. The left bowl of an 'o' might not be symmetrical in shape nor match in thickness the right bowl. Thin serifs and strokes might "drop out" or "break up" (disappear) from some letters. Because of round-off error, some letters might be a pixel shorter or taller than others so their Y-axis maxima and minima would not align and the letters would seem to bounce up and down along the crucial baseline (the imaginary line on which most letters seem to sit visually) — or along the x-line or along the capital height line. Reading would therefore seem like driving on a bumpy road. At high resolutions, around 900 to 1,200 dots per inch or more, such variations in height and thickness seem to be imperceptible to most readers, but at medium resolutions of 300 dots per inch, and at screen resolutions around 72 dots per inch, such variations were evident and often objectionable. Murch & Beaton investigated the effects of resolution and addressability on perceived image quality of CRT displays.

G. Murch & R. J. Beaton, "Matching display resolution and addressability to human visual capacity," *Displays* 9(1), 1988.

Differences in calculations of resolutions needed for high quality text rendering and the actual resolutions used in high quality graphic arts printing may be due, at least in part, to the technical meaning of "resolution." Resolution is commonly used for the number of pixels or dots per inch (or millimeter), but that is technically "addressability," the spacing density of the raster grid. Strictly speaking, "resolution" means the dimensions of the dots or pixels written on the raster grid. Hence, the use of resolution, including in this essay, actually means addressability. On CRT

displays and printers, dot size is at least 1.5 times larger, and often 2 or 3 times larger, than addressability. To print a solid color, a dot size substantially larger than dot spacing enables the dots to merge into an evenly solid area instead of creating a stipple or stripe effect. On LCD displays, physical pixel sizes are fixed and resolution is very close to addressability, although many displays can simulate coarser resolutions by combining pixels.

[Note 25]

For much of the 20th century, vision research on typography was to determine the practical legibility of typefaces, with consistent findings that size of type was more important than typeface style. More recent research has used type forms as visual objects to investigate fundamental aspects of visual perception. These fundamental studies did not influence the Font Wars, but have improved our understanding of how type forms are visually recognized and processed, individually and in reading. As reading migrates from print to digital screens on smart phones and e-readers, such studies increase in importance. The pioneering 1968 paper by Campbell & Robson was about sinusoidal gratings, not type, but inspired later psychophysical studies that did involve type. Here are just a few of many interesting and important papers.

F. W. Campbell & J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *The Journal of Physiology*, 197(3), 1968.

A visual example and discussion of the contrast-sensitivity function is at:

http://ohzawa-lab.bpe.es.osaka-u.ac.jp/ohzawa-lab/izumi/CSF/A_JG_RobsonCSFchart.html

including:

<http://ohzawa-lab.bpe.es.osaka-u.ac.jp/ohzawa-lab/izumi/CSF/CSFchart640x480.gif>

A short list of later papers involving letter recognition and reading include:

A. P. Ginsburg, "Visual Information Processing Based on Spatial Filters Constrained by Biological Data" (No. AMRL-TR-78-129-VOL-1/2). Air Force Aerospace Medical Research Lab Wright-Patterson AFB, OH, 1978.

R. A. Morris, "Image processing aspects of type," In *Document Manipulation and Typography: Proceedings of the International Conference on Electronic Publishing, Document Manipulation and Typography*, Cambridge University Press, 1988.

R. A. Morris, "Classification of digital typefaces using spectral signatures," *Pattern Recognition*, 25(8), 1992, pp. 869–876.

J. A. Solomon & D. G. Pelli, "The visual filter mediating letter identification" *Nature*, 369(6479), 1994, p. 395–397.

Majaj et al. found that the spatial frequency channels used by the human visual system to recognize letters are different depending on letter size. The authors give a dramatic visual

illustration of the effect.

N. J. Majaj, D. G. Pelli, P. Kurshan, & M. Palomares, “The role of spatial frequency channels in letter identification,” *Vision Research*, 42(9), 2002, 1165–1184.

Majaj et al. is discussed by Gordon Legge in relation to the spatial frequency model of reading:

G. E. Legge, *Psychophysics of Reading in Normal and Low Vision*. Mahwah, NJ: Lawrence Erlbaum Associates, 2007, pp. 59–65, 117–122, 123–125, 132.

Related views by typographers and type designers include:

C. Bigelow, “Form, pattern, & texture in the typographic image,” *Fine Print* 15(1), 1989.

C. Bigelow & K. Holmes, “Science and history behind the design of Lucida,” *TUGboat*, Volume 39, No. 3, 2018. <https://tug.org/TUGboat/tb39-3/tb123bigelow-lucida.pdf>

[Note 26]

Typographic structural constraints are of two general kinds, alignments and elements. For example, take a sans-serif, lowercase letter n. It has two vertical stem elements which should have roughly equal thickness in pixels. At high resolutions, a slight difference in thickness may be visually unnoticeable, and there may be small differences by design, but at low resolutions, a difference of a pixel is noticeable. Also, the n has two terminals at the baseline that must align at the same pixel boundary, not only within the letter n but also with the baseline terminals of letters like h, l, m, x, the horizontal base stroke of z, etc. And these should appear to align with the curved bottoms of o, d, p, q, as well as the bottoms of v and w. The n also has a terminal at the x-height, as does i, j, m, the diagonal strokes of v, w, x, y, and z, and so on. The n has an arch connecting its two stems. At high resolutions, the arch is somewhat higher than the flat terminal of the n at the x-line (in a sans-serif design). If the small additional vertical extension is carefully adjusted, the arch will visually have the same height as the curved upper bowls of the letter o, as well as b, d, e, p, q, r, and s. In a serifed typeface, the frequently replicated serif shapes may be constrained to be identical or at least very similar in form.

The first programmed solution to the regularity problem was devised around 1979 by Peter Karow at URW and incorporated in the Ikarus software system. It was implemented as a module called “PA” or “Passe” (a German word meaning “fitting”). A human operator, assisted by pattern recognition software, analyzed the metrics and distributions of letter features, such as dimensions of stem widths, letter heights and widths, base alignments, and so on, essentially preparing a histogram of clusters of features. That analysis was then programmatically applied to the Ikarus character outlines prior to scan conversion, adjusting the Hermite splines of character outlines to conform to the metrics of the output raster grid, thereby inducing symmetry in the resulting bit patterns.

P. Karow, “Two decades of typographic research at URW: a retrospective,” In *International Conference on Raster Imaging and Digital Typography*, 1998, pp. 265–280.

Karow stated he did not patent his pioneering work on font regularization, deeming it too obvious for a patent.

A recent personal memoir and history of URW by Peter Karow is:

P. Karow, *Pioneering Years: History of URW, Part I*, J. & P. Dougherty tr., URW Publishing Department, Hamburg: 2019

In Ikarus analysis, and also in later hinted font technologies, like Adobe Type 1 fonts, spline-outlined letters were digitized according to certain rules for on-curve spline points (“knots”) placement. Points were placed at the extremes of outlines in the X and Y axes, to regulate baselines, x-heights, the maximum swell of bowls (like the curved parts of o), and several other features. As a few examples, the corner points on the flat bottom of a letter ‘l’ would be marked on the baseline, as would the y-axis minimum extent of the bottom curve of a o, that may slightly under-hang the baseline. Similarly, spline curve points would be placed at the top curve of an o and the top of the arch of an n. In addition, parallel edges of stems would be marked. Ikarus also enabled duplication of features like serifs, to regularize repeated parts. When character outline data was regularized in this manner, segments of contours could be stretched or compressed slightly to coincide with integer (or sometimes for curve extremes, fractional) values of the output raster grid, thus maintaining symmetries that otherwise would be lost in raw scan conversion.

Roger Hersch provided a general overview, analysis, and explanation of character contour curves, grid-fitting, and rasterization, with illustrations and bibliography. Hersch compared PostScript Type 1 horizontal and vertical “banding” zones for grid fitting to more general TrueType instructions.

R. D. Hersch, “Font rasterization: the state of the art,” In *From object modelling to advanced visual communication*, Berlin, Heidelberg: Springer, 1994, pp. 274–296.

Beat Stamm explains the concept and implementation of an interactive editor for grid-fitting instructions.

B. Stamm, “Visual TrueType: A graphical method for authoring font intelligence,” in *International Conference on Raster Imaging and Digital Typography*, Berlin, Heidelberg: Springer, 1998, pp. 77–92.

It should be mentioned here that progress in research in digital font technology was advanced by several academic conferences in the 1980s and 1990s, including the “Computer and the Hand in Type Design” seminar in 1983, the “Raster Imaging and Digital Typography” conferences in 1989 and 1991, the “Visual and Technical Aspects of Type” conference in 1993, the Project DIDOT conferences, and others.

In a 2012 account of the development of PostScript, John Warnock stated that the hinting concept was implemented after the founding of Adobe.

J. Warnock, “The origins of PostScript,” *IEEE Annals of the History of Computing*, vol. 40(3), 2018, pp. 68–76. By autumn of 1983, hinted control of letter features had been implemented in the Type 1 font interpreter.

Adobe did not apply for a patent on its hinted font technology, believing that the idea was so simple that even with a patent, work-arounds would be found. Adobe did disclose confidential descriptions of PostScript to potential customers in 1983 and printed a PostScript manual in spring of 1984. These disclosures did not reveal details of Adobe’s font technology but did spur others to develop font regularizing font techniques for inclusion in printer and typesetter controllers.

Selected Patents on Font Regularization

Thomas Hawkins and Compugraphic filed for a patent on “Intellifont” in July 1984, granted in 1987: United States Patent 4,675,830, T. B. Hawkins, inventor.

Working at Bitstream, Phillip Apley and others filed for a patent on “Automated bitmap character generation from outlines,” in 1986, granted in 1988: United States Patent 4,785,391. This was followed by two more patents assigned to Bitstream: “Outline-to-bitmap character generator,” filed in 1988, granted in 1990: United States Patent 4,959,801; and “Method and apparatus for conversion of outline characters to bitmap characters,” filed in 1990, granted in 1992: United States Patent 5,099, 435.

[Note 27]

The differences between the “hinting” of Adobe’s Type 1 font technology versus that of Folio’s F3 technology were termed “declarative” versus “procedural”. Adobe’s declarative hinting resulted in compact fonts with relatively small file sizes, in which the hints were terse “declarations” of marked spline knots. Adobe’s complex interpreter-rasterizer turned declared hints into adjusted outlines and rasterized the results. Folio’s procedural hinting produced more complex fonts with more data because each character included its own hint program code. The in-font grid adjustment code enabled the F3 adjuster and rasterizer to be fast, compact, and easily portable. Hence, there were trade-offs between font file size, rasterizer complexity, and processing speed.

[Note 28]

A concise history of Knuth’s development of TeX and Metafont is in:

B. Beeton, K. Berry, & D. Walden, “TeX: A Branch in Desktop Publishing Evolution, Part 1,” *IEEE Annals of the History of Computing*, 40(3), 2018, pp. 78–93. <https://tug.org/pubs/annals-18-19/>

By design, Metafont produces bitmap fonts, not outline fonts. Hence, characters produced with Metafont did not need the “hints” or “instructions” that PostScript and TrueType outline characters needed for aesthetic rendering, as Metafont’s output is already at the level of pixels. (It is possible to tune the bitmaps for the eventual intended output device.) Knuth early on observed a rounding problem in which pixel patterns on curves depend on the phase where a digitized pen

stroke, or abstractly, a curve arc, intersects a discrete raster grid. If the pen boundary falls coincides exactly with an integer grid line, a pimple-like pixel can stick out of a run of pixels, but if the stroke boundary falls slightly shy of the grid, a noticeably flat run of pixels occurs. A visually better pixel pattern is produced when the stroke boundary falls midway between grid lines [17].

Another outcome of Metafont generating bitmap fonts is that it fell to other groups to eventually create alternative implementations of Computer Modern, as outline formats, particularly PostScript Type 1, became prevalent in printing. Around 1990, Type 1 fonts corresponding to each of the Computer Modern fonts created by Knuth, and several adjunct fonts of symbols, were produced in a cooperative venture between Blue Sky Research (Barry Smith, Doug Henderson), which had developed TeX software for the Macintosh; Y&Y Inc. (Berthold and Blenda Horn), which had developed TeX software for Windows along with methods of hinting fonts in Type 1 format; and Projective Solutions (Ian Morrison, Henry Pinkham), which developed mathematical-based tools for spline editing and conversion, and font building.

The Computer Modern Type 1 fonts were created partly by generating extremely high-resolution bitmaps and tracing the results, partly by inspecting the splines that Metafont created in response to the program's instructions, and partly by manual editing. These Type 1 fonts are still the form in which Computer Modern is most widely used today, though TrueType and OpenType versions of Computer Modern, with many extensions to character repertoire, have also been created and are also commonly seen.

Knuth's doctoral student, John Hobby, created a sibling to Metafont, named MetaPost, which is a Metafont-like drawing tool that outputs PostScript graphics code instead of bitmap fonts (Hobby, 1989, 1992). Its PostScript output has also been used to create outline fonts (a much less laborious task than starting from Metafont's output, as PostScript is not pixel-oriented). Perhaps the most notable achievements in this regard are by a group of typographer-programmers in Poland known as the GUST e-foundry; their fonts and articles can be found online via <http://www.gust.org.pl/projects/e-foundry>.

[Note 29]

The term “non-Latin” was common in the typeface catalogs of typesetting equipment manufacturers, as a convenient albeit imprecise category. “Non-Latin” is admittedly an inexact and Latin-centric term that lumps together many different writing systems and scripts with different histories, linguistic structures and graphical expressions. One hopes a more accurate and appropriate term will be agreed upon. Most of this essay and notes have concerned technical inventions focused on Latin typography, but the history of technical East Asian typography, including Chinese, Japanese, Korean, and traditional Vietnamese writing systems deserves more study.

A recent history on the Chinese typewriter in relation to Chinese culture discusses a subset of the broader study of the typography of East Asian writing systems.

T. S. Mullaney, *The Chinese Typewriter: A History*, Cambridge: The MIT Press, 2018.

[Note 30]

A familiar example of glyph substitution in English text involves combinations of the letter ‘f’ followed by f, i, or l, as in ff, fi, fl, fj, ffi, ffl. These combinations can be visually awkward if the upper arm of f collides with the dot of the following i or j or the ascender of l. This can sometimes be seen in texts composed in Times Roman (or Times New Roman) without use of ligatures. Traditional typesetters therefore cast such combinations as single forms called “ligatures,” because the letters were “tied” together. Typesetters, either manual or operating composing machines, would replace, for example, a string of letters such as f f i with a single ligatured ffi character. In Latin typefaces, ligature substitution has aesthetic value, although there are common design modifications of f and i that make ligatures and glyph substitutions unnecessary.

In contrast, glyph substitutions are obligatory in traditional Arabic scripts in which letters have shape and joinery variations based on context. An Arabic letter form may differ depending on its position in a word — at the beginning, in the middle, at the end, or isolated — and there are other variations depending on the script style. Tom Milo presented a tutorial on Arabic script forms and encodings at the Unicode Conference in 2002.

T. Milo, “Arabic Script Tutorial,” 29th Internationalization and Unicode Conference, San Francisco, CA: March 2006.

A recent history of the typography of Arabic script in the 20th century provides scholarly, technical, and aesthetic analysis.

T. Nemeth, *Arabic Type-Making in the Machine Age: The influence of Technology on the Form of Arabic Type, 1908–1993*, Leiden: Brill, 2017.

The writing systems of India, such as Devanagari, Bengali, and others, often collectively termed “Indic Scripts,” also have obligatory glyph substitutions and ligatures, as do several other scripts of Southeast Asia. For the most part, traditional analog typography and early digital typography were not able to render the full range of typographic expression of these and related writing systems. Digital typography provides methods for implementing these scripts. For example, making Devanagari fonts in OpenType is explained here:

<https://docs.microsoft.com/en-us/typography/script-development/devanagari>

As an example of glyph substitution in a Latin alphabet font, the ornamental script typeface Apple Chancery, designed by Kris Holmes and produced by Apple, demonstrates the capabilities of the original Apple GX line layout technology. Apple Chancery contains different glyphs for initial, medial, and final forms of letters in words and for letters beginning or ending lines, as well as for isolated letters. The font also contains a large complement of ligatures, including many for common English letter pairs such as Th and th.

Apple GX line layout technology is described in United States Patent 5,416,898, “Apparatus and method for generating textual lines layouts,” Inventors: Opstad, David G. & Mader, Eric R., May 16, 1995. Assignee: Apple Computer, Inc.

A history and comparison of TrueType GX Line Layout to OpenType, “Comparing GX Line Layout and OpenType layout,” was written by Dave Opstad, the principal architect for TrueType GX Line Layout at Apple. The document is posted on-line at:

<http://mirror.informatimago.com/next/developer.apple.com/fonts/WhitePapers/GXvsOTLayout.html>

The current OpenType specification is version 1.8.3:
<https://docs.microsoft.com/en-us/typography/opentype/spec/>

The current Unicode Standard is version 12.1.0:
<http://unicode.org/standard/standard.html>

Unicode currently supports 154 different scripts that can be used to write an even greater number of languages.

Interpolatory font formats.

In the 1990s, two new, intriguing, interpolatory font formats were developed and released by font warring companies. One was released in 1991 by Adobe and named “Multiple Masters,” and the other was released by Apple in 1995 and named “TrueType GX Variations.” Both were based, in different ways, on the idea of interpolating character shapes pioneered by Peter Karow in the Ikarus system in the 1970s. Ikarus interpolation was based on a function familiar to mathematicians as a mapping of each contour point of one character to a corresponding point of another character, provided that the mapping is “one-to-one and onto” (or similar meaning in different nomenclatures). Each character needed to have the same number of contour points, and each contour point had to map to a corresponding point of the same kind, such as a corner, tangent, or curve. Interpolation was often used in Ikarus to produce intermediate weights of a font by interpolating between a light and a bold weight.

Adobe’s Multiple Master system brought font interpolation to graphic designers and font users, who could adjust the amount of interpolation between different forms of particular fonts, including between different weights, different widths, different “optical masters” designed for different output sizes, and other variations. Apple’s TrueType GX Variations offered similar capabilities, but handled the interpolations differently, basing them on a basic middle form and interpolating toward extremes. These competing technologies never became opposing weapons in the font wars because both failed in the marketplace. Although interpolation became a common and useful feature in font design tools, it did not catch on with users. In Adobe’s case, Multiple Master fonts were expensive and time consuming to design and produce, and could not compete in price or variety with the flood of cheap, pirated fonts let loose by Adobe’s 1990 disclosure of the Type 1 font format. Moreover, to follow through on its OpenType alliance with Microsoft, Adobe began shifting engineering and font development to OpenType in 1997. Development of Multiple Master fonts ended in 1998, and shipments ended by 2000. At Apple, only one TrueType GX Variation font was shipped with TrueType GX, around 1995. Despite marketplace failures of the particular implementations, the idea of interpolatory type designs

lived on and two decades later found new proponents in technology and font firms. A new version derived from the Apple invention was revived in 2016 through cooperation between Adobe, Apple, Google, Microsoft, and Monotype. It is named OpenType Font Variations and sometimes called OpenType Variable fonts or simply Variable Fonts. Its overview and specification is incorporated into the OpenType 1.8 specification of 2018:

<https://docs.microsoft.com/en-us/typography/opentype/spec/otvaroverview>

[Note 31]

Note 31 has evolved into a self-contained article that will be published in 2020 in *TUGboat*, the journal of the TeX Users Group (<http://tug.org/tugboat/contents.html>). Here we have a preprint of the article: see [fontwars-note31.pdf](#)

[Note 32]

Font Wars “Note 31” draws parallels between the mid-15th century invention of typography and the late 20th century invention of personal computing. It is based on Paul M. Romer’s macroeconomic theory of the relationship between technological ideas and economic growth. Note 31 also illuminates aspects of the “dematerialization” of fonts in the history of typographic technology, in light of Romer’s theory.

P. M. Romer, “Endogenous technological change,” *Journal of Political Economy*, 98(5, Part 2), 1990, pp. S71-S102. Cited as (Romer, 1990).

This note 32 takes a microeconomic look at the fates of font technologies in the Font Wars, when several font technologies competed but only a few were ultimately successful. The path from a good idea to a successful product is often fraught with risk and uncertainty, and success in a business venture is contingent upon many factors. In the Font Wars, four major factors appeared to be platform dominance, promotion of a standard, effective technology, and executive commitment.

1. PLATFORM DOMINANCE

A computing platform may comprise hardware, operating system, other software environments, or combinations thereof that support systems, applications, services, and other software. Indicators of platform dominance may include the number of users and network interactions, the number of developers and associated interactions, the amount of prestige of the platform, or other indices as may be defined for a given sector of an industry or market. In 1989, at the time of the Font Wars announcements, several firms had some platform dominance. These included Microsoft, Apple, Adobe Systems, Sun Microsystems, and possibly a few others.

Microsoft. Microsoft was by far the leading vendor of computer operating systems; its MS-DOS, OS/2, and Windows operating systems dominated the personal computing market. Foreseeing the importance of font technology for future versions of OS/2 and Windows, Microsoft investigated several outline and hinted font technologies in 1988-1989. Microsoft’s 1989 announcement that it would adopt Apple’s “Royal” font technology boosted that technology to potential hegemony. (In the following paragraphs, the “Royal” technology is also

called by its final release name, “TrueType.”)

Apple. Although Apple had less than 8% of Microsoft’s operating system market share, it enjoyed greater prestige as a leader in innovation, ease-of-use, and style. In desktop publishing, Apple was the dominant computing platform. It had introduced desktop publishing to the mass market in 1985 with the Macintosh personal computer and the LaserWriter printer incorporating Adobe’s PostScript graphics software. In 1989, the Macintosh platform had the most popular graphics and publishing applications, including Aldus PageMaker, Adobe Illustrator, and QuarkXPress. Microsoft’s dominance in operating systems and Apple’s prestige in desktop publishing made a strong alliance supporting Royal/TrueType font technology.

Sun Microsystems. In 1989, Sun was the dominant firm in the workstation market, especially in workstations using Unix or Unix-like systems. Sun was a leader in supporting and developing industry standards. It had developed its own PostScript-like screen display system, NeWS, and had acquired the Folio corporation with its F3 font technology in 1988. Some observers supposed that the combination of those two technologies would lead to Sun’s dominance in imaging and font technology standards among workstations.

Adobe. Adobe created the PostScript page description language (PDL) and its Type 1 hinted font technology, which were launched in the Apple LaserWriter printer in 1985. By 1989, Adobe had also licensed the PostScript PDL and its font technology to several other firms, including Linotype, thus creating a large, interactive network of users of Adobe technology and developers of applications based on Adobe technology, such as Aldus and Quark. Hence, Adobe’s PostScript had become a virtual platform in the graphic arts, printing, and publishing industry, including desktop publishing and personal computing using text and graphics.

Other firms. Most other developers of font technology lacked sufficient numbers of uses and developers to constitute a dominant platform, and thus needed to partner with a dominant firm in order to have a chance of success in font technology.

URW was a long-established, innovative R&D software developer for the typesetting industry. URW was the first software firm to develop cubic-curve outline font technology and software to fit outlines to digital raster grids. URW’s Ikarus system for digital font development was licensed by several major manufacturers of digital typesetting systems, including Compugraphic, Linotype, Monotype, and Autologic. URW’s “Nimbus” hinted font technology was developed in the late 1980s-1990s. As a small firm, compared to Microsoft or Apple, URW needed to partner with a larger firm for its font technology to become a success. Microsoft considered a version of URW Nimbus, called “Nimbus Q,” before deciding on Apple’s Royal.

Compugraphic was the leading manufacturer of digital typesetting equipment in the 1980s; its patented “Intellifont” technology had been derived from URW technology in the mid-1980s. Microsoft considered adoption of Compugraphic’s Intellifont before choosing Royal TrueType, and Hewlett Packard did adopt Intellifont technology for its LaserJet III printers.

Folio was a small start-up corporation that in 1986-1989 developed the F3 font technology with automatic “hinting” and a compact font rasterizer portable to most operating systems. Microsoft

and Sun both investigated Folio's F3 technology, and Sun acquired Folio in 1988.

2. PROMOTION OF A TECHNOLOGY STANDARD

The Font Wars included a battle over a "standard" for font technology, based on the premise that the more users there are of a standard, the more that a platform supporting the standard gains dominance through the network of users and developers of the standard. At the 1989 Seybold Seminar, there was strident debate over whether there should be only one standard, or two, or three.

An often repeated remark attributed to Andrew Tanenbaum (but sometimes also to Grace Murray Hopper) is:

"The nice thing about standards is that there are so many to choose from."

Adobe. Based on experience at Xerox PARC, where the "Press" and "Interpress" digital imaging and print standards had been proposed and developed, Adobe promoted PostScript and its font technology as a "standard." This marketing claim benefitted from a lack of contemporary competition. Xerox's "Interpress" standard had not yet been commercialized in shipping printers and did not specify a particular font technology. Imagen's Document Description Language (DDL) with its hinted font technology had been a competitor to PostScript but did not advance to market after Hewlett Packard cancelled its license with Imagen in 1987. Thus, Adobe's PostScript font technology constituted a *de facto* outline font standard until the early 1990s. The PostScript Type 1 font standard benefitted Adobe synergistically. As inventor and licensor of the Type 1 font format, Adobe gained monopoly profits from sales of Type 1 fonts produced by PostScript licensees as well as by Adobe itself. Moreover, through trade secrecy, Adobe was able to exclude competitors and thus dominate the platform to sustain its monopoly profits.

Microsoft-Apple. Beginning in 1987, Apple developed its Royal font technology to gain financial and entrepreneurial independence from Adobe and to control its core technology. Apple's financial motivation has often been cited as a reason to develop its own font technology, but a deeper motivation was Apple's strategy to "control its own destiny." In practice, that meant that Apple wanted to own and control its font technology, seen as a vital success factor, in order to control and drive innovation on its own platform.

Microsoft investigated several font technologies but did not settle on one until the summer of 1989, when Apple and Microsoft ignored their on-going lawsuit over user-interface "look and feel," and formed a strategic partnership based on Apple's Royal font technology. Their joint announcement at the September 1989 Seybold Seminar claimed that the Royal font technology was superior to Adobe's and would become the personal computing industry standard. Such was the dominance and prestige of their partnership that the Royal font technology was assumed to be the main competitor to Adobe technology, more than a year before Royal, renamed TrueType, was launched in spring 1991.

Sun Microsystems. Sun's installed base was far smaller than those of Microsoft or Apple, but in the high performance workstation market, and especially workstations using Unix-based operating systems, Sun was the dominant firm in high-end business and technical computing

platforms. Sun had been founded on a strategy of adopting technical standards and was a leader in adopting, supporting and developing standards like the TCP/IP internet protocol and the Network File System. Support of standards helped Sun achieve platform dominance. By 1990, at least one industry expert observed that Sun was nearing “critical mass” in establishing an industry standard with its NeWS graphical screen display system combined with F3 font technology.

Hewlett Packard. Hewlett Packard (HP) was the computer industry’s largest laser printer manufacturer, with around two billion dollars in annual sales by late 1989. HP adopted Compugraphic Intellifont for its LaserJet III printers in 1990, but even HP’s dominant printer position was not sufficient to outweigh the overall platform dominance of Microsoft and Apple combined.

3. EFFECTIVE TECHNOLOGY

Many of the battles in the Font Wars were over alleged technical superiorities of the competing font technologies. Among the criteria were: speed of rasterization; compactness of font file size; code size and portability of rasterizer; aesthetic quality and fidelity of output; efficiency of input of font digitization, hinting, and production.

Later, exponential progress in computer hardware obviated most of the performance issues that had been red-hot topics in the Font Wars. The seemingly inexorable workings of Moore’s Law increased processor speeds and memory capacities by orders of magnitude, while chip prices fell at similar rates. Laser printer resolutions increased more slowly than chip performances, but printer resolutions doubled to 600 dots per inch in less than a decade. Higher resolution coupled with rendering enhancements reduced the need for, and importance of, grid-adjustment hinting. On computer screens, resolutions doubled, and on small devices like tablet computers and smart phones, doubled again. Those high resolutions combined with anti-aliasing techniques generated visual type quality that seemed equivalent to traditional print. By the 21st century, the font technologies that had survived the Font Wars were more or less equally effective, albeit with lingering technical differences.

4. EXECUTIVE COMMITMENT

Apart from platform dominance and effective technology, a crucial factor for success in the Font Wars was corporate executive commitment to a technology strategy, communication of that strategy to the managers and engineers developing the technology, and continuing support of the technology in order to achieve platform success and standardization. Such strategies required considerable investment and therefore involved taking substantial risks, because failure was a possible outcome in the highly competitive Font Wars.

Apple. In 1987, Jean-Louis Gassée, Apple Vice President, initiated the development of Apple’s font technology. It was not a plan to derive monopoly profits from the invention itself but a strategy to avoid paying royalties to Adobe on PostScript font and printer technology and, more deeply, to control Apple’s core technology to enable future, profitable innovations without interference from outside technology licensors. Apple invested around four years in development with several engineers and project managers assigned to the effort. Apple also invited outside developers of rival font technologies to share information, ostensibly so Apple could design its

technology to facilitate translations from the outside technologies into Apple's forthcoming font technology. Several developers accepted the invitation.

To further support its font technology development, Apple engaged expert font consultants and sent teams of engineers to Germany to confer with URW, a long-experienced and respected innovator of font technology for the typesetting industry. Apple commissioned URW to write software from URW's Ikarus font format to TrueType, which URW then incorporated into its Ikarus font development software for the Macintosh platform. For end users, Apple produced a set of core fonts to compete with Adobe's original basic set, and in addition commissioned TrueType redesigns of four of Apple's bitmap fonts first shipped with the original Macintosh system.

To further promote TrueType, Apple hosted a well-attended TrueType Font "Jamboree" for font and applications developers in December 1989.

A tactical feature, or more precisely the absence of a feature, was that TrueType fonts were not encrypted. Its font format was openly specified so that third party developers could produce TrueType fonts without a license from Apple. Apple did, however, patent certain aspects of the TrueType rasterizer, though that did not have appreciable effect on fonts. Apple's corporate strategy - cessation of royalty payments to Adobe and resumption of control over core technology - was not based on expectation of profits from fonts, so font encryption was not needed. The problem of font piracy in an open format would directly impact font designers, developers, and vendors, but not Apple.

TrueType font technology proved to be technically successful, and Apple achieved industry-wide distribution by partnering with Microsoft.

Adobe. Between 1985 and 1989, Adobe's PostScript Type 1 font technology had become a successful virtual platform in the graphic arts and publishing industry. In the summer of 1989, however, Adobe's executives recognized the threat from Apple's Royal font technology, then still in development. Adobe shifted engineers to the development of Adobe Type Manager (ATM), designed as a stand-alone Type 1 font rasterizer for Macintosh, to be marketed directly to Macintosh end users, potentially circumventing Apple's rival font technology. Though the announcement of the alliance of Microsoft with Apple and Royal technology appeared to be a surprise at the September 1989 Seybold Seminar, Adobe was already planning to release ATM for Macintosh by the end of 1989, and did so. In 1990, ATM became a well-reviewed and popular product.

Hence, with executive foresight and commitment, Adobe preceded Apple's font technology by more than a year. TrueType did not ship until it was integrated with Macintosh System 7 released in May 1991.

Adobe launched ATM for Microsoft Windows in October of 1990, a year and a half before Microsoft shipped TrueType with Windows version 3.1 in April 1992. Adobe's success in selling the ATM application directly to end-users, together with Adobe's prior success in selling Adobe Illustrator to end-users, showed that Adobe could be a successful application developer

and vendor.

The possibility of a greater supply of unencrypted TrueType fonts for Macintosh and Windows platforms posed a threat to the *de facto* standard of PostScript Type 1. Adobe countered this threat as well. Following the Microsoft-Apple announcement of Royal at the 1989 Seybold Seminar, Adobe's CEO John Warnock announced that Adobe would "open" its secret Type 1 font format, to protect the PostScript standard. Warnock reportedly stated that, "Adobe PostScript is so important to the publishing industry that I'm not going to let it fail." That affirmation of support for PostScript by the co-founder and CEO of Adobe was to reassure developers, users, and licensees of Adobe technology (ComputerWorld, Sept. 25, 1989). Far-reaching consequences of Adobe's publication of the Type 1 format are discussed in previous Note 31.

Microsoft. Though Microsoft was dominant among computing platforms, its strategy was to acquire a font technology instead of developing its own. It investigated several rival technologies from both technical and business perspectives. Though it ultimately agreed to license Apple's TrueType, Microsoft recognized that it would be playing catch-up in technology and font quality to Adobe's well established font technology and fonts. During the two and a half year time lag between the Seybold announcement in 1989 and Microsoft's shipment of TrueType with Windows 3.1 in the spring of 1992, Adobe released its ATM font technology application for Windows in autumn 1990, well in advance of outline font technology on Microsoft's own platform.

With the support of top executives, Microsoft was determined to make its TrueType core set of fonts at least equal in quality to the similar fonts from Adobe or preferably superior, especially on screen displays, where the fine-tuning capabilities of TrueType had an advantage over PostScript Type 1, though at a greater cost of labor and production time.

Taking what seemed to be a "spare no expense" approach, Microsoft licensed from the Monotype corporation a series of fonts to emulate Adobe's original LaserWriter and LaserWriter Plus fonts. Monotype provided font outline data from an Ikarus system licensed from URW, but programming TrueType grid-fitting instructions was a non-trivial task. Microsoft tackled this problem with licensed technology and human labor. To convert Monotype's Ikarus outline data to the quadratic B-splines of TrueType, Microsoft licensed Spline Lab, a digital curve conversion tool created by a small firm, Projective Solutions, founded by mathematicians Henry Pinkham and Ian Morrison. To program TrueType grid-fitting instructions, Microsoft licensed a hinting tool, Type Man, from Type Solutions, a small firm founded by Sampo Kaasila. To code the instructions for the TrueType font programs, Microsoft assembled a team of more than a dozen designers, some employed by Monotype and some free-lancers with experience in font hinting. Microsoft also licensed the source code for TypeMan and later used that as a starting point for the Visual TrueType hinting tool, offering it for free to TrueType font developers. Microsoft also hired outside consultants to review printer output and screen display fonts for quality assurance.

When Microsoft released TrueType in Windows 3.1, it bundled a core set of fonts competitive in style and metrics with Adobe's core set, as well as an original "Wingdings" font of symbols and icons. At the same time, Microsoft released a "Microsoft TrueType Font Pack for Windows" that

contained forty-four fonts, more than half of them original designs for digital typography. Thus, Microsoft's 1992 font releases not only matched Adobe's standard font sets in PostScript printers but also challenged Adobe's "Originals" digital font program by issuing approximately the same number of original digital type designs as Adobe and distributing them in far greater volumes, as of that year.

Sun Microsystems. From 1989 to 1992, Sun Microsystems' F3 (Folio Font Format) appeared to be a viable contender in the Font Wars. Developed by the small Folio corporation beginning in 1987, the character contours of F3 were composed of compact, high-resolution conic curves that could be rasterized faster than the Bézier curves of PostScript fonts while being more compact than the quadratic B-splines of TrueType fonts. In 1988, both Sun Microsystems and Microsoft showed interest in Folio, which was eventually acquired by Sun, then the dominant Unix-based workstation manufacturer. The Folio-Sun F3 technology comprised two main parts. One was an advanced, automatic hinting system called "TypeMaker" that solved the bottlenecks of Type 1 and TrueType formats - the substantial amounts of human designer labor needed to produce fonts with grid-fitting hints, also called "instructions." The automatic hinting made F3 attractive to major font vendors like Linotype and Monotype, which agreed to license the technology.

The second part of F3 was called "TypeScaler." This was the interpreter that scaled and rasterized the F3 hint code to produce type for screens and printers. TypeScaler was written in the C language and was easily portable to other computer platforms, including Unix systems, without incurring the overhead of a page description language such as PostScript.

At the time it purchased Folio, Sun had in-house graphic technologies, including conic-based graphics and the NeWS network windowing system developed by James Gosling. Sun's dominance in the workstation market led some industry observers to predict that F3 would become a font technology standard in the Unix world, as had other standards that Sun adopted or supported and that had contributed to Sun's platform success.

Sun created a group called SunPics to market display and printer products, which were launched in 1990, the same year that Adobe released ATM. TrueType was not released until 1991 by Apple and 1992 by Microsoft, so Sun had a slight head start on those larger firms.

M. Marshall, "Sun Brings PostScript to Non-PostScript Printers," InfoWorld, October 1, 1990.

After the creation of SunPics and its products, however, Sun began to reduce support of F3. A few Sun executives began to discourage the licensing of F3 technology to major corporations, and, contrary to initial expectations, F3 did not get integrated into Solaris, Sun's version of Unix. Sun's support of F3 for font vendors was curtailed, as was support of F3 for third party font tools developers.

Sun's apparently inexplicable suppression of F3 bewildered F3 customers, licensors, and font vendors, until October 1992, when Sun and Adobe announced that Sun would license Display PostScript and PostScript print software. A report of the deal described a year of secret negotiations between a few Sun executives and Adobe:

CBR Staff Writer, “Display PostScript is Sun’s Way Out of Its NeWS Bind,” Computer Business Review Online, 27 October 1992.

According to this CBR report, SunPics’ managers had been kept unaware of the secret negotiations until shortly before the October 1992 announcement. SunPics’ managers objected to the deal not only because they had been excluded from the negotiations but also because Sun would be paying Adobe for screen display and font technology equivalent to what Sun had already developed or acquired, and because Sun would be relinquishing control of core technology.

SunPics’ objections relate to principles of Paul M. Romer’s economics model discussed in Note 31. In Romer’s model, inventors can benefit from research and development that generates successful products from which the inventors can gain monopoly profits. Sun had already made substantial investments in imaging and technology, namely NeWS and F3, which provided functionality equivalent to, or in some respects superior to, PostScript imaging and fonts.

By using its own font technology, Sun would not need to pay an outside firm for the technology. Additionally, use of its own technology would enable Sun to continue to innovate technologies that relied on fonts and graphics, without needing permissions and payments to an outside firm. These were the main issues that had motivated Apple to develop, and Microsoft to adopt, TrueType: elimination of payments to an outside vendor as well as technological independence and freedom of innovation, more important in the competitive, fast-innovating computer industry.

Some years after Sun’s 1992 announcement of the Display PostScript deal, Sun abandoned Adobe Display PostScript and Type 1 font technology, but in the absence of a committed executive strategy for font technology, Sun didn’t revive F3. Instead, Sun adopted a succession of outside font technologies, discovered problems with them and abandoned them, until eventually settling on TrueType in the late 1990s. By then, the Folio inventors had left Sun, and F3 was unrecoverable as a viable technology. It is possible that, for historical interest, the once innovative F3 software may still reside in archives somewhere at Oracle, which acquired Sun in 2010.

[End of Part 2 notes]