Screen Typography

Applying Lessons of Print to Computer Displays

Paul Kahn
Krzysztof Lenk

This article appeared originally in the
Seybold Report on Desktop Publishing
Volume 7, Number 3

Dynamic Diagrams, Inc.

12 Bassett St.
Providence, RI 02903
T 401.331 2014
F 401.331 2015
E-mail Paul_Kahn@DynamicDiagrams.com
http://www.DynamicDiagrams.com

Walk into any major bookstore and you will see a tremendous amount of information presented in the form of ink on paper, the paper cut into pages, the pages bound in folios. Compact, portable, cost-effective, sporting a simple-to-learn common interface, the printed book has been adapted to transmit all or part of every form of human communication. At the root of the book is writing, verbal communication in the form of text, represented in the form of type.

The art of European typography goes back to the 15th century. It grew from a manuscript tradition in which letter form and page composition had evolved for centuries. In the past five hundred years we have continued to develop techniques for presenting text on a printed page. These techniques serve many ends—aesthetical, functional, commercial—but ultimately the information represented in type must be easy and pleasant to read.

The History of Type on the Screen

Text has been present on digital computers from their inception in the 1940s in the form of programming languages (input) and program result (output). For several decades the representation of text was quite limited, initially to single case since that is all that was necessary to represent programs and output. Support for upper- and lower-case letters came only after the introduction of 8-bit computers in the 1960s. Finally there were enough bits to represent 128 characters, but the letters themselves were presented on the screen as a pattern of dots in cells of a uniform size. Text came in and out of computers on a teletype (a typewriter that communicated over a distance) or terminal (from the sense of that word meaning the end of an electrical connection). The computer could accept, store and retrieve a code for each letter, which could be presented on the terminal screen as a single dot pattern in the form of 80 such patterns to the line and 24 lines to the screen.

Computers that supported a graphical user interface on bitmap displays were created at Xerox PARC in the 1970s. These computers changed the way text was represented on the computer screen in a dramatic and decisive way. Once all the points on the screen could be addressed, a computer could present text in a wide variety of forms. Formatted text appeared on the screen representing different letter designs, sizes and weights. By the mid-1980s the Xerox Star, the Apple Lisa and finally the Apple Macintosh had brought type to the computer screen in a fashion that has defined the computing environment we work in today.

As the computer and publishing industries today focus on the integration of sound and video into electronic publications, the convergence of computing and telecommunications, and the interconnectivity of enormous computer networks, we are presented with a major problem. After a decade of reading formatted text on the screen, we are not very happy about what we see. People do not want to read lengthy passages of text on a computer screen.

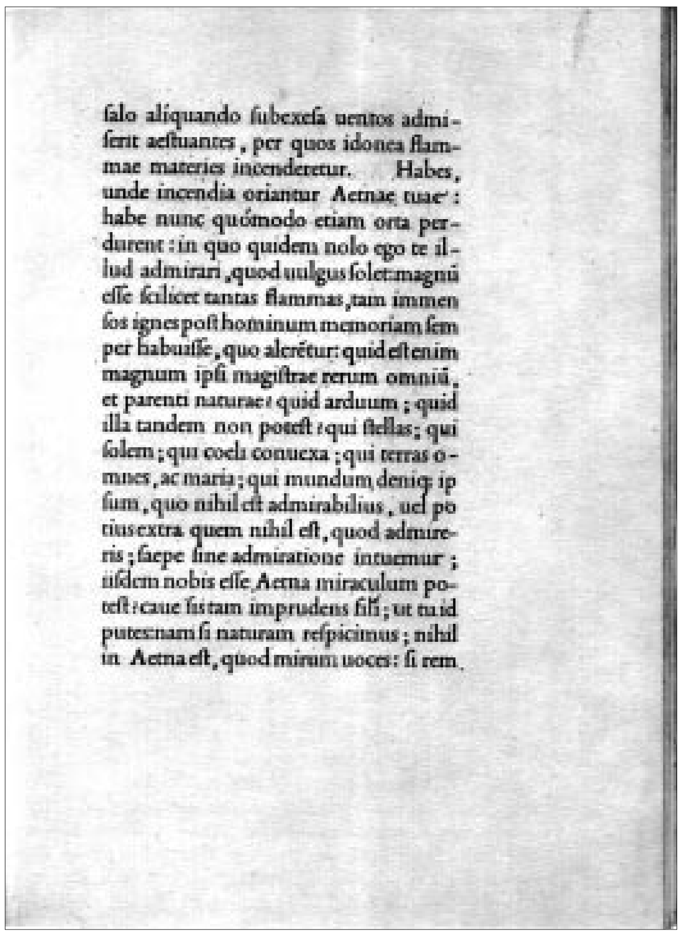
It doesn’t have to be that way. Even within the limits of today’s technology, there are better and worse ways to put text on the screen. In this article we hope to define legibility on the screen, at least those aspects that relate to typography, and propose an agenda for improving the current state of affairs.

The Screen and the Printer

The situation of text on the computer screen today can be understood only if we briefly review the history of computers and their output devices: the typewriter, the typesetter and the laser printer.

Someone discovered that a computer could drive a typewriter and word processing was born. Several fortunes have been won and lost since the magnetic card typewriter was introduced. The word processor is still very much with us today in the form of programs designed to put text on a sheet of paper. The conceptual model of these programs is still the typewriter with tab settings, ruler and sliders defining the left and right margins, even though many of the people who use them today have never used an electric typewriter or its mechanical predecessor. In a word processor the text on the screen must represent the text on the line, but not necessarily the text on the page. Calculating the number of characters on a line was very simple because the width of characters on a typewriter was simple. All characters were of the same width.

Figure 1
A page from Pietro Bembo's De Aetna printed by Aldus Manutius, Venice 1495-1496. Note the use of white space in the margins to offset the solid block of text.


ſalo aliquando ſubexeſa uentos admi-
ſerit aeſtuantes, per quos idonea flam-
mae materies incenderetur. Habes,
unde incendia oriantur Aetnae tuae:
habe nunc quómodo etiam orta per-
durent: in quo quidem nolo ego te il-
lud admirari, quod uulgus ſolet; magnũ
eſſe ſcilicet tantas flammas, tam immen
ſos ignes poſt hominum memoriam ſem
per habuiſſe, quo alerētur: quid eſt enim
magnum ipſi magiſtrae rerum omniũ,
et parenti naturae? quid arduum; quid
illa tandem non poteſt? qui ſtellas; qui
ſolem; qui coeli conuexa; qui terras o-
mnes, ac maria; qui mundum deniq; ip
ſum, quo nihil eſt admirabilius, uel po
tius extra quem nihil eſt, quod admire-
ris; ſaepe ſine admiratione intuemur;
iiſdem nobis eſſe Aetna miraculum po-
teſt? caue ſis tam imprudens fili; ut tu id
putes: nam ſi naturam reſpicimus; nihil
in Aetna eſt, quod mirum uoces: ſi rem

Someone discovered that a computer could drive a typesetting machine and computer typesetting was born. Fortunes rose and fell in this world as well as the typesetting machines evolved from phototype to digital imaging. In one form or another computer typesetting programs are the source of nearly all ink on paper that we read today. The conceptual model of these programs is the typeset galley, that is, the column of text exposed onto photographic paper or film in a specific font, size and column width. The text on the screen of a typesetting program had to represent the quantity of text on the galley line along with the codes informing the typesetting machine about what font, size and column width to use. Calculating the number of characters to the line was complex. Character widths varied for each font and size, as did the size of wordspaces and the space between characters in a word. Before ending a line decisions had to be made about whether and where to hyphenate the last word.

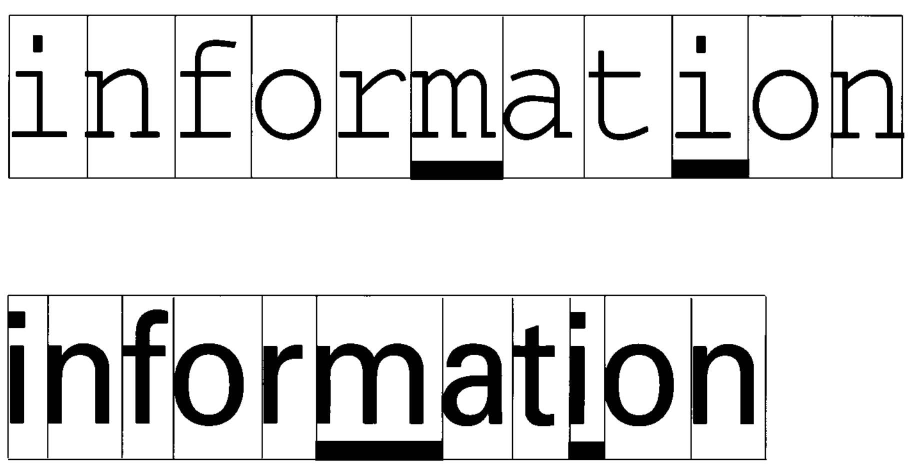
Someone discovered that a computer could drive a xerographic copier and the laser printer was born. The laser printer started out imitating the way text came off a typewriter but with the combination of the Apple Macintosh, LaserWriter and Adobe PostScript the model changed. The laser printer looked like a typesetter, with various fonts, sizes and weights of type. The laser printer set pages on paper rather than galleys on film. The text on the screen needed to be an approximation of the text on the page with all the variables of a typeset line. It could only approximate the page, however, because the resolution of the screen was very different from the resolution of the laser printer.

Someone discovered things we could do on the computer screen that we could not do on the printer. Hypertext technology allows us to follow links from one piece of text or graphic information to another. Information retrieval allows us to create a list of all occurrences of a word or phrase and then travel to each one. Links between programs allow us to illustrate the information being presented dynamically. Multimedia technology integrates sound, animation and video. The information on the screen is no longer a surrogate for the real information on the printed page. The surface of the computer screen is the page. The screen fonts are no longer the approximation of the type our readers will see—these screen fonts are the type we must use to communicate our ideas.

The common theme in the first three stories is that text on the computer screen is intended to represent what text will look like on the printer. There are plenty of things we can do on the printer that we can only suggest on the screen, not the least of which is draw type fonts from 4 to 16 to 45 times the resolution available on the typical computer screen.

The theme of the last story is quite different. The relationship between text on the screen and on the printer has changed. The computer industry seems to have missed the significance of that difference.

Figure 2
Comparison of monospace font (Courier) and variable space font (Univers). Note the typically wide variation in space between the m and i in Univers.



How We Read

Typography is an applied art. A typographer is communicating the information in a text through the selection and arrangement of type on a surface. Typography is also a visual art, subject to all rules of visual syntax we usually associate with other visual arts such as painting, sculpture and photography. The arrangement of type on a surface communicates through our nonverbal system of perception as well as our verbal sense of what the words "say." The treatment and placement of text can create powerful word pictures, as in the masterful creations of Paul Rand—logos for IBM and UPS which have become a part of our daily visual landscape. The communication of ideas occurs as we read the text in both a verbal and nonverbal sense.

The techniques of typography evolved from applied practice, not from scientific theory. There is, however, a body of psychological research that helps to explain why some typography works better than others.

Miles Tinker, whose published research on reading spanned five decades, points out that "the printed page contains no meanings but only symbols which stand for meaning" *(Basis for Effective Writing,* 1965). We convert those symbols into meaning through a series of eye movements. We read from left to right (speaking of European languages) perceiving a combination of word groups and individual letter forms, mixed with regressive movements from right to left, until we perceive the end of a line. At the end of the line we sweep our eye left to locate the beginning of the next line and begin the process again. There is a hint of this psychophysiology in the English meaning of verse (a line of poetry) which comes from the Latin versus, a turning, in this case of the eye's focus.

A great deal of what makes text easy and pleasant or awkward and difficult to read is the relationship between the way text is presented and the human visual system. Herbert Spencer, now retired from the Royal College of Art in London, summed it up in this way:

> *Perception in normal reading is by word wholes, and the recognition of the importance of the total word form has led some writers to assume that only the general word shape or external outline is important in reading perception. But although word shape may make a valuable contribution it is necessary also to recognize that words consist of internal patterns as well as outlines and that details of internal pattern may provide cues which are essential to accurate perception. Tinker has stressed the distinction between 'total word shape,' the bare outline of a word, and 'total word structure.' The distinction is an important one. (Spencer, The Visible Word, p. 20)*

The font designer must consider a number of variables when creating the letter shapes we read from. Charles Bigelow, a contemporary type designer, has analyzed the readability of a font in terms of six common features: size, weight, contrast, spatial frequency, proportion and differentiation. Each of these features is balanced in relation to the others when designing a font.

The font size of what we read in print is relatively constant. Most book, newspaper or magazine text is set in 9- or 10-point type. This size is well tuned to the capacity of the human visual system. However, when we try to apply this

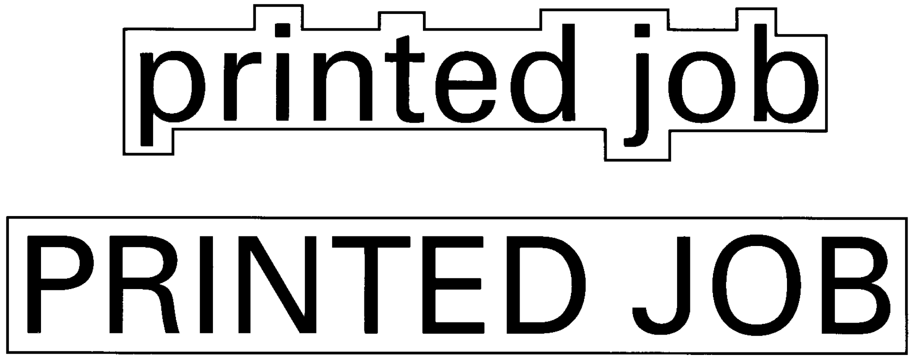


Figure 3
A comparison of the word shape in upper and lower case versus all capital letters. Our perception of word shape largely relies on the rise and fall of ascender and descender characters, which are absent from capital letters. This is why blocks of text set in all capital letters (such as software licenses) are difficult to read.

lesson of printed text to the computer screen there is a fundamental mismatch: at current screen resolutions of 72 dots per inch there are not enough pixels to represent well the letter forms in 9- or 10-point type. There are also not enough pixels to represent adequately many of the individual characteristics that distinguish the hundreds of digital fonts designed to produce text on typesetters and laser printers. Many common characteristics suffer. Diagonal strokes that make up Z, X, V, W and Y cannot be rendered as smooth lines and become “jaggies.” The serifs that cap the ends of all strokes in many typefaces are either compressed or exaggerated. The variation of thickness that defines the visual rhythm of the letter form can be lost with only one or two pixels to define a stroke.

> *The practical evidence suggests that today’s screen resolutions of 72 lines per inch are at least one or two decimal orders of magnitude too low to produce text of optimum visual quality. (Bigelow, “Font Design for Personal Workstations,” Jan. 1985 Byte, p. 256)*

This would suggest that a larger point size than what we use in print, say 12- to 14-point type, would be better suited to presenting type on the screen. But Bigelow is quick to point out that simply adding a few pixels to an analog design will not solve the problem.

> *The lesson here is that traditional analog typefaces cannot be imitated and jaggies cannot be eliminated on today’s display screens. The only practical solution is to design screen fonts within the limitations of the available raster system, to optimize the font’s features to the mechanisms of the human visual system, and to make sure these features conform to the familiar historical principles of letter design. (Bigelow, “Font Design for Personal Workstations,” Ibid.)*

Bigelow has addressed these problems with several of his own type designs. His type foundry, Bigelow & Holmes, has produced Lucida, a family of fonts designed for high-resolution typesetters, medium-resolution laser printers and lower-resolution computer screens (see “The Design of Lucida,” *EP 86,* pp. 1-17). Examples of Lucida are available under X Window/Motif, Sun’s Open Look, as TrueType fonts for Microsoft Windows, and as Adobe Type 1 fonts for the Macintosh.

We also have fonts such as Chicago, Monaco, Geneva and New York on the Macintosh, which were designed as screen fonts alone. These fonts were designed to present file names and menu items, individual words and phrases within the general user interface, and are not well suited for presenting blocks of text. They lack the visual rhythm needed to support sustained reading. Far more interesting for the present discussion are Arial and Times New Roman, the TrueType fonts shipped with Microsoft Windows 3.1. These fonts have been optimized for screen presentation of blocks of text as well as legibility on higher-resolution printing devices.

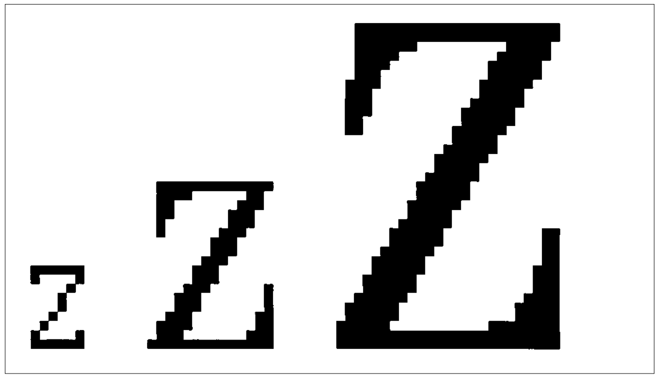


Figure 4
Screen bitmaps for Z in New Century Schoolbook in 12-point, 24-point and 48-point (enlarged). Note how the serif is the same thickness as the horizontal at 12-point and only suggests a curve at larger sizes. Also, the variation i thickness between the horizontal and diagonal strokes appears only in the larger sizes.

Balancing the Black and the White

The typographer is concerned with visual and verbal balance. He is an information architect, to borrow Richard Saul Wurman's phrase, giving shape to a space within which the verbal meaning of a text can be perceived and explored.

There is an old saying among German typographers: "Typography is the art of using black to express whiteness." Every letter has an edge and most have internal spaces, called counter forms, such as the circle inside the O. Every word ends in a space or a punctuation mark. Every line has a beginning and an end, a top and a bottom. Text is a combination of the black positive shape of the letter and the white negative space that defines the black.

The balance of black and white is just as important on the screen as it is on the printed page. As fundamental as the black letter shapes are, it is the white negative spaces of counter forms, intercharacter and interword spaces, line spacing and margins that support or undermine the type designer's best efforts. The typographer must have control of all these elements to work his art.

The letter shape of a screen font can be produced in either of two ways: by displaying a predefined bitmap for each character in each font in each size, or by computing the bitmap on the fly to best match an outline for each character. We will call the former method "hand-tuned" because these predefined bitmaps are nearly always edited by typographic designers to obtain the best results. We will call the latter method "scalable" because the bitmap must be scaled to fit the convergence of the outline and the screen resolution.

A good deal of attention has been given to improving letter shapes on the computer screen in recent years. With the introduction of Adobe Type Manager (ATM) and TrueType to rasterize many font bitmaps from a single font description, the quality of bitmaps at arbitrary sizes has improved. To say that they have improved is not to suggest that any 8-point, 11-point and 17-point bitmap of any font drawn by either ATM or TrueType rasterizer is necessarily good, only that it is better than the algorithmic distortion of the hand-tuned 12- or 14- point bitmap that we had to live with before these products were introduced. However, it is important to understand that we will not solve the legibility problems by improving letter shape alone. Equal attention must be given to the other ways in which the typographer controls both the positive and the negative space that make up the text we are presenting on the screen.

Figure 5
The same word set normal, tight and very tight. The gray area on the right indicates the cumulative space saved. This space is taken from between characters, drawing together vertical strokes in the a, i and n and overexposing the counter forms.

The Margins

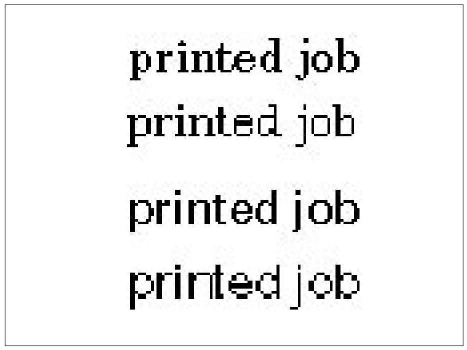
The edges of a block of text are defined by margins. Margins are not necessarily empty—they include the header and footer area where folio and other navigation information may appear. But whatever appears in the margin, it is clearly separated from the space that contains the main text block. On the printed page there are certain conventional proportions of text column to margins that support legibility. Tinker analyzed the pages of 400 textbooks and found an average of 52.8% text area to 47.2% margin. He noted that most readers are not aware of what he called the “50 per cent rule.” Most subjects in his study thought print represented 60-75% of the page. (Tinker, *Legibility of Print,* 1963, p. 111-112)

The size and placement of margins in book and magazine design is determined by the reader’s visual field. This consists not of one but of two pages spread on a flat surface. Jan Tschichold, a leading theorist and practitioner of modern typography, worked out a geometric relationship he felt was the best for book pages based on the proportions he found in manuscript books. The height of the type area is equal to the width of the page with the margins divided into 2:3:4:6 proportions, as shown in the diagram at upper right on page 7. (Tschichold, *The Form of the Book*, 1991, p. 44)

The equivalent of the page for the screen designer depends on how the application software manages the screen. It is either the entire monitor or the window within which the application lives. This presents a series of rectangular areas from which the text block must be separated.

Like the subjects in Tinker’s study, most computer users neither see nor value margins in relation to text. While we have never had to read text from a book or magazine where the text fills the entire page, we have become numb to this factor on the computer screen. There is a black margin on most monitors between where the monitor glass meets the plastic or metal frame and where the rasterizer begins to draw on the glass. Under Unix or dos, the system

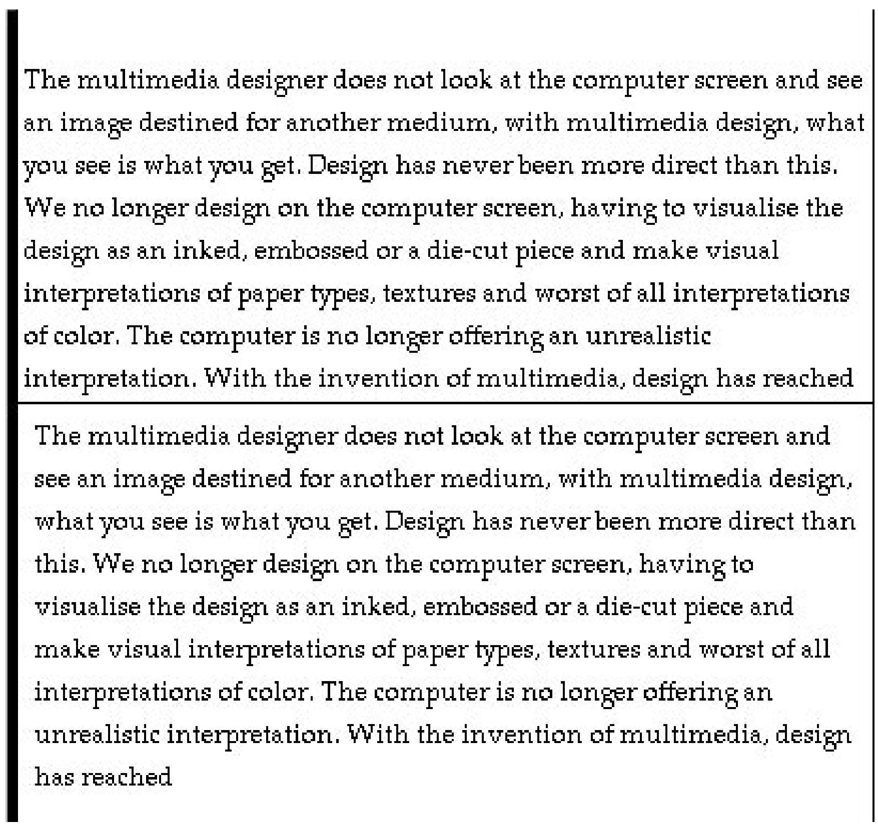
Figure 6
Screen bitmaps (enlarged) in 17-point New Century Schoolbook with ATM (first line) and without ATM, Univers with ATM (third line) and without ATM (fourth line). Note how ATM makes sure the thickness of the vertical stroke in the p, d and j are the same.



prompts and any text typed to the monitor begin at the left-most pixel on the monitor. Most word processing programs include a default where the left edge of the text block is placed a few pixels away from the edge of the window in which the text is being edited. The “real” left margin, that half inch to one inch of white space we expect to see on the printed page, is stored as a value in an instruction to the printer. The assumption is that showing the left margin on the screen wastes space that can otherwise be used to display the text on the line.

We call this “word processor syndrome” and it is not limited to word processing programs alone. Text fields in HyperCard 2.1, one of the leading toolkits for multimedia publications, come with two possible left margin settings: a standard three pixels or “wide margins” of eight pixels! The award-winning Grolier Electronic Encyclopedia presented the text of articles in windows with three pixels of margin for several years. Such compression of the left margin is a serious obstacle to legible text.

Figure 7
The margins in scrolling text fields in HyperCard 2.1. The margin in the upper field in three pixels (default). The margin in the lower field is eight pixels (wide margins).



The purpose of the margin is to expose the edges of the text block. This exposed space is necessary to express the separation of the text block from other text blocks and from the window/page border. It is also necessary to aid the eye in locating the beginning of the line as it sweeps from the end of one line to the beginning of the next.

Both the indent off the left margin and the space between paragraphs are also an important structural feature of the text block. These negative spaces express the structural grouping of text to the eye at a level that precedes the reading of words. Variations off the straight line of the left margin guide the eye through the reading process. A carefully chosen series of indents, built up from a proportion of the text point size, can express the level of information in each text block. Text that contains a list or series of items is easiest to read when the individual items are marked with numbers or nonverbal markers. However, a second margin is needed to create a sharp edge for the text in the list, separating the text from the marker, e.g., the "hanging indent."

Paragraphs can also be separated without indenting by adding space between blocks. When the white area between blocks is greater than the interline space, this space joins the left and right margin to frame individual elements of the text block. In a similar vein, a title or heading can be joined to one text block and separated from another by variations in this space above and below the line. However, for these techniques to be effective the designer must have control of the white space that defines indents and line space. Few multimedia authoring programs provide such controls.

When the screen is a series of overlapping windows, the margins within each window become increasingly important elements. We dramatically increased the amount of white space around text in our design work with the IRIS Intermedia hypertext system to help expose text blocks when they appear in the context of multiple window stacks.

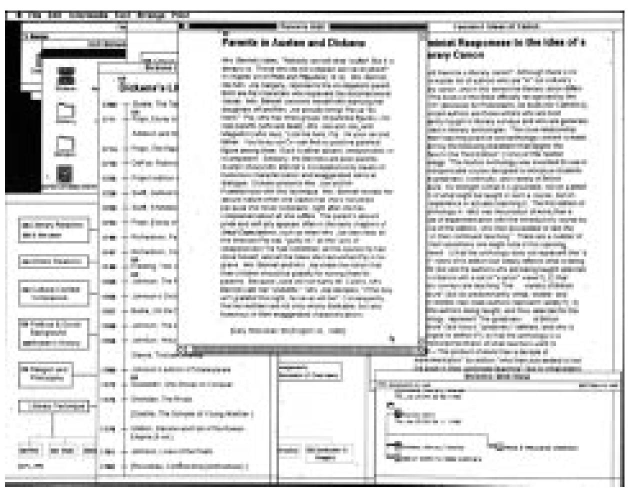


Figure 8
A sample from the pioneering hypertext system, IRIS Intermedia, circa 1989. Note the wide margins used to set the text block apart in the window stack.

The Column of Text

> *"A column of type has a tranquil appearance and is easy and pleasant to read when a unity of set-width, word spacing, line length, and leading has been achieved." (Ruedi Rüegg, Basic Typography: Design with Letters, p. 39)*

The typesetter creates a line of type, but the typographer creates a column of text. We read text in columns, as a continuous shape. As Rüegg points out, the column of text should be as calm and even as we can make it. Its unity and balance should help the eye, not confuse it or slow it down.

Experience with print has shown that a column of text should be between 40 and 60 characters wide for optimal legibility. This figure must be balanced with other factors such as the size of type and the width of the type area (page or window area minus the margin area). When a longer line length is necessary, the interline space can be increased. This increase in white space between the lines will help the eye find the following line as it sweeps farther in the right-to-left motion.

The reason for the column width limit is the human visual system. The eye grows tired of scanning text without pause. Sixty characters is a desirable upper limit of this scanning process before the break of a return sweep is needed to reset the attention. This is also the upper limit before which the eye has an increasingly difficult time locating the start of the next line. When the eye is unsure, the reader must regress and check context to match the last word of one line with the first word of the next. This slows and inhibits the reading process.

This column-width rule is why magazines and newsletters divide a standard letter-size page into two or three columns of text. On the screen the use of multiple columns of text is rare due to the large point size needed to present clear letter forms and the size of most monitors.

An interesting and instructive example of this rule in action is the layout of Voyager's Expanded Books series. The layout was designed to fill the screen of the Apple Macintosh PowerBook, which is wider and shallower than most Macintosh monitors. The designers at Voyager established a text area that averages 40 to 50 characters per line using 12-point Palatino type. The wide left and right margins are put to use supporting annotations and user interface elements.

Publications we have designed using Electronic Book Technologies' DynaText also follow this rule, though users are free to resize the text block to suit their own preferences. DynaText presents both a table of contents view of a

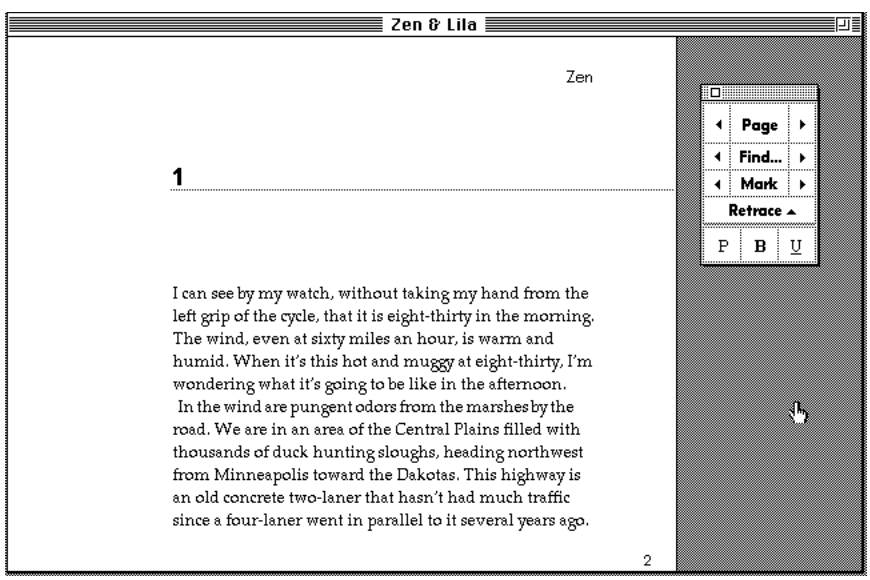


Figure 9
An example screen from Voyager's Expanded Books for the Apple PowerBook. Voyager provides ample margins, which are nice annotations but also improve the overall legibility of the main text.

book’s structure and a view of the main text in two panes of a single window. The preferred orientation and proportion of these panes depends largely on the size of the monitor. The vga standard of 640∞480 pixels on most Microsoft Windows machines is best suited to a vertical split with a 2:3 ratio of table of contents to text. Unix workstations such as the Silicon Graphics Indigo and the Sun Sparc-2 have larger monitors. Text on these monitors is better suited to a horizontal split, with the table of contents above the main text view taking up only part of the screen.

Just as the margin space defines the edges of the column of text, the interline space defines the shape of each word. This is still sometimes called leading, referring to the strips of lead that were placed between lines of metal type to increase the visual space between them. Word shape is largely defined by the ascenders and descenders, those horizontal and diagonal strokes or lack of them that provide distinction between letters such as a, q, g and d; n and h; v and y; and t and f.

If a column of text is “set solid,” that is, the point size of the text and the line spacing are the same, there will be no space between the descenders of the first line and the ascenders of the second line. In print it is common to add 2 points of leading between each line, so that a line of 9-point type will be set on 11 points of line spacing. Most text applications add one or two points (in the form of single pixels) of space between lines.

While this follows the general print rule, it is often not enough space to distinguish type on the screen clearly. When tops and bottoms of letters come within one or two pixels, they confuse the eye and inhibit reading. In the publications we design, such as the help system for Interleaf’s WorldView product, we add a few points of interline spacing beyond the default to lighten the density of the text block. We find that while this decreases the number of lines of

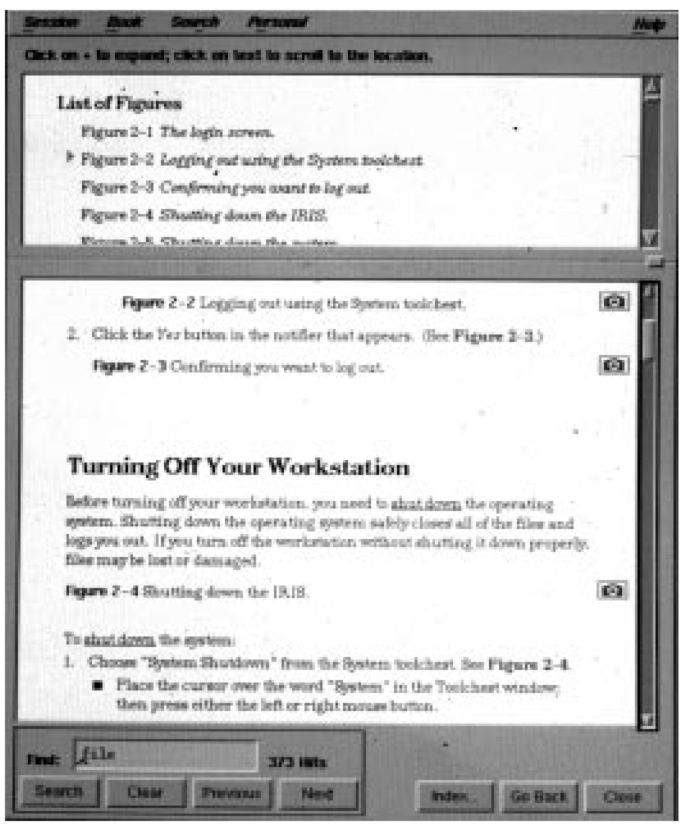


Figure 10
An example of a book in IRIS InSight (a version of DynaText) for the Silicon Graphics workstation. Here, the contents view is at the top instead of on the left.

text within a window, that loss of text is made up for by an increase in the legibility. The Voyager Expanded Books design also increases the line spacing over the default, setting 12-point Palatino on 17 points of space.

The space between lines must be controlled in pixels or points. The typewriter model of “space, space and a half, double space” is woefully inadequate for controlling this important feature, yet this is what the designer has to work with in the current version of Asymetrix ToolBook. The designer working with text in Macromedia Director has no control of line spacing at all.

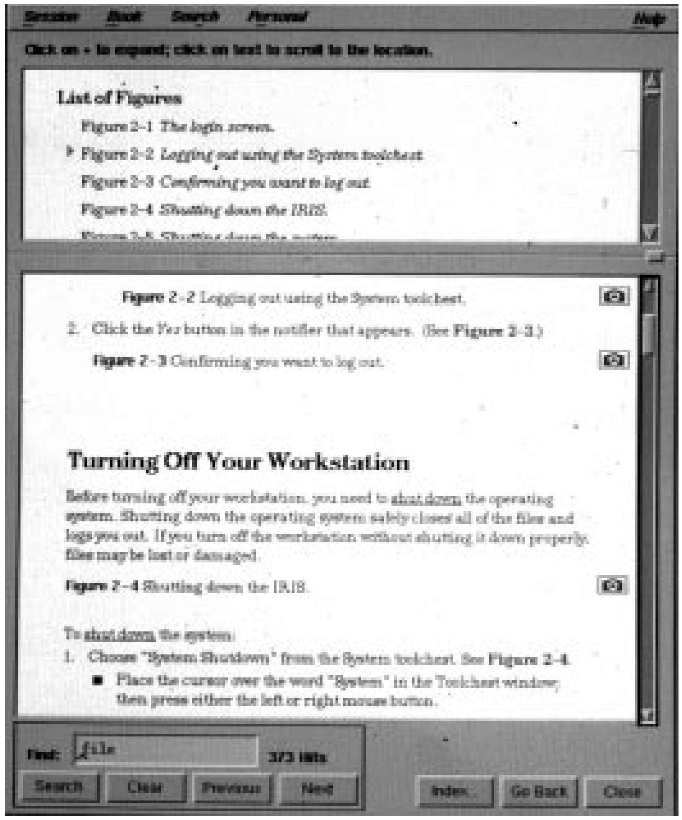


Figure 11
An example of a book in IRIS InSight (a version of DynaText) for the Silicon Graphics workstation. Here, the contents view is at the top instead of on the left.

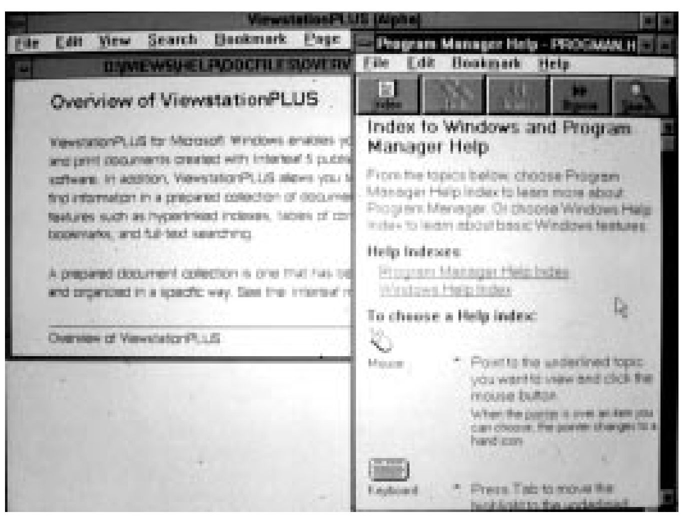


Figure 12
An example of a book in IRIS InSight (a version of DynaText) for the Silicon Graphics workstation. Here, the contents view is at the top instead of on the left.

The WYSIWYG Battleground

The legibility of text is also defined by the space between characters and between words. Essentially the shape of words is created by the relationship between these two kinds of space.

The space between characters should appear to be a regular interval to the eye. The relationship between positive and negative space is not regular across character shapes, however, so to create this optical evenness some careful variation in intercharacter space between specific letter pairs is needed. The space between two letters composed of vertical strokes such as i and f should be different from the space between a vertical and a round letter such as f and e. The left side bearing of the e can fit into the negative space defined by the right side of the f. Information on such “kerning pairs” is found in the outline description of both Adobe Type 1 and TrueType fonts. However, this information is rarely used by programs to control the space between character bitmaps on the screen.

The generally irregular intercharacter spacing in most screen fonts is as much a contributor to poor legibility as jaggies and poor letter forms. This irregular spacing is the result of two things: the rounding up or down in width of the character bitmap to fit the character outline and the difference between the width of the bitmap in pixels and the set width of the character on a higher resolution printer. The battle for correspondence between what we see on the computer screen and what we get on a printer is fought in the spaces between characters on the screen.

The problem is most dramatic in programs that try both to set a line of text for arbitrary printer output and to preview the text column on a line-for-line basis. A program such as Aldus PageMaker positions the left edge of each character bitmap on the pixel closest to the position of the character on the target printer with no regard for the effect this has on intercharacter spacing on the screen.

Programs not so concerned with typesetting use the standard character width values of each character in the font to calculate where to draw each bitmap. This provides at least one pixel between characters at sizes above 12-point but still results in irregular spacing in most fonts.

Left-justified text can be set either ragged (with an uneven right edge) or justified (with an even right edge). In most book, magazine and newspaper applications, justified text is preferred because it creates a more even column shape. The uneven distribution of white space on the right edge of a ragged column of text can be visually distracting. Far more distracting, however, is an uneven use of interword spacing to justify a line. (See “H&J: The Art of Composition” in *The Seybold Report on Desktop Publishing,* Vol. 2, No. 8, April 4, 1988, for an exposition of how hyphenation and justification was, is and should be done. Nothing of significance has changed in the ensuing five years.)

An oft-practiced rule of thumb is that the ideal space between words should be the equivalent to the lowercase e in the type font. To justify a line of text, the space between words must vary in order to redistribute the white space between the end of the line and the right margin. If this is done without hyphenation, the interword spaces will clump together into unintended vertical patterns of negative space, called rivers because they seem to flow through the text block. Even with hyphenation, a well-justified text column will require variations in interword spacing and an adjustment to intercharacter spacing as well. Without hyphenation, the creation of a legible justified text column is simply impossible.

Eliyezer Kohen, an engineering scientist at Microsoft who has done a great deal of work with TrueType, has made an interesting suggestion as to how an application might better utilize the character space information in the font description to create evenly spaced text at screen resolutions. Kohen suggests that all line-break calculations be done in what he calls the design space. Each character in TrueType is specified in 2,048 design units per em. (Adobe Type 1 fonts are specified in 1,000 design units to the em.) These specifications include left and right side bearings, character shape and set width. An application should read all this information for each character, scale the result for output on an arbitrarily high-resolution printer, say capable of addressing 32,000 dots per inch, and calculate the appropriate line break. The application should then look at the actual number of pixels in the sum of all character widths and the actual number of pixels in the text column space. If the column is 600 pixels wide and the sum of the set width of all characters is 592 pixels, the eight-pixel difference should be distributed among the word spaces to justify the line. If the set width is 612 pixels, the 12-pixel difference should be taken away from the word spaces up to a certain tolerance, and then taken away from the space between specific character pairs.

Kohen also explained that his group at Microsoft put a great deal of time and effort into "hinting" the character space information for the three TrueType fonts that shipped with every copy of Microsoft Windows 3.1: Arial, Times New Roman and Courier. The result of their work is very impressive, particularly in the case of Arial at both very small sizes such as 8- and 9-point and at common screen sizes such as 12- and 14-point. At the small sizes all left side bearings have been set to zero and all right side bearings set to one pixel, producing a minimum uniform spacing at these small sizes where the intercharacter boundaries collapse in most fonts. Unfortunately this careful coding of set widths and spacing is not a characteristic of TrueType fonts in general.

None of the current applications available from Microsoft support Kohen's suggestion on how to render a line of text on the screen, so we have not seen whether his method is a sound one. It would seem that such a usage of TrueType technology would be particularly appropriate for the Microsoft Viewer product, which is used by many publishers to produce MPC publications. Applications such as Interleaf's WorldView perform similar calculations on font metrics in Interleaf's own page description language to distribute space on a pixel-by-pixel basis into wordspaces.

At present, no software vendor offers support for hyphenation and justification of a column of text specifically tuned for display on the computer screen. The desktop publishing market has brought very sophisticated hyphenation and justification capabilities to the Macintosh and Microsoft Windows machines, but this same facility is entirely lacking in hypertext and multimedia software. Such a facility does not come for free, in a computational sense. Calculating possible hyphenation points throughout a large document each time a user resizes a window could certainly slow down the pace at which the text is rendered on the screen, but there is no doubt that it can be done.

Typographic Distinction

Within a block of text we often want to distinguish a specific word or phrase from all the others. Print typography has developed the convention of using a bold or italic version of the text font to set apart words in this fashion. Such variation in type style adds voice to a line, which sounds a little heavier if it appears in black.

Mixing regular, bold and italic is the equivalent of adding color in one-color printing. When two-, three- or four-color printing is used, different color inks can distinguish words or blocks of text from one another. The considerable cost of adding two or more colors on a printed page is rarely justified by using color in text (we are not, of course, talking about text for advertising or display purposes here). The contrast of black ink on white paper is optimal for legibility and any other color will somewhat decrease legibility. Using several colors can be distracting and requires great care. Richard Saul Wurman's innovative Access Press travel guides are that rare example of a publication that successfully makes extensive use of color in text. In addition to the usual uses of bold to set off proper names and italic to indicate titles or non-English words, black, blue, green and red are used to separate blocks of text about culture, hotels, shopping and restaurants.

The lack of matching bold or italic faces on the typewriter led to the convention of underlining words for emphasis, since this was an effect that could be achieved by overstriking a word with the underline key. In place of bold many typists put on the caps lock and put a WORD in all caps to make it louder or more important.

We have the same needs for typographic distinction on the computer screen. Our options are limited by several new factors, however.

Using italic, which is a slanting of all letter shapes, decreases legibility of most screen bitmaps to such an extent that it is unusable on the screen as a method of emphasis. The addition of separate font outlines and bitmaps for italic (as well as bold and bold italic) versions of fonts now gives us a "true" italic rather than an algorithmic distortion of the regular bitmap, and this is an improvement. But even in the most legible of screen fonts, the italic version is far more difficult to read.

Emboldening a screen font usually involves doubling or otherwise increasing the number of pixels used to render the letter shape. This fills much of the counter space in small font sizes, but often works well at sizes such as 12- and 14-point. Scaling algorithms produce an interesting "false bold" effect at the point where the basic stem width of characters jumps from one to two pixels. This occurs somewhere between 14- and 18-point, depending on the font. This

Figure 13
Underlining in Help for Microsoft Windows 3.1. Note how the solid underline cuts through the descenders, making it difficult to see the letter g. The dotted underline also interferes with legibility, but to a lesser degree.

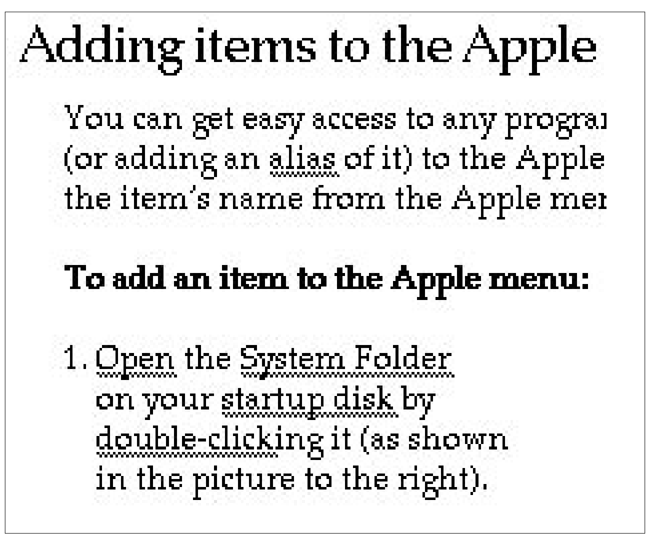
doubling makes regular text appear to be bold in relation to medium text of the same font in smaller sizes. As a result, heading text set in 18- to 24-point sizes in screen fonts rarely needs the addition of bold to create emphasis.

Color is cheap on the screen compared to color on the printed page. Given a monitor that supports 8-bit color, it costs no more, either in terms of equipment or disk space, to display text in a mixture of colors. As a result, this form of emphasis is used frequently in screen design, particularly under Microsoft Windows where monochrome monitors are rare. Legibility is based on the contrast of the color of the letter form with the color of the background, so whatever color is chosen it must contrast clearly with the background (yellow text on a white background is the clearest bad example). The high saturation values of some "pure" colors are very hard on the eye. When choosing color for text we tend to tone down the saturation value whenever possible, choosing maroon rather than pure red, slate blue rather than pure blue, and so forth. Color distinction in text often needs to be reinforced with a shift to bold type because many letter forms in small sizes are only one pixel wide. The visual system is more likely to pick up a difference in color when the color area is at least two pixels.

Color creates hierarchy. Certain colors come forward in relation to black while others recede. This is particularly noticeable in screen type where a word set in gray or a light blue will "fall away" from the eye in relation to the rest of a text block set in black while a dark red word will stand out and come forward in the visual hierarchy. On a computer with very fine color control, such as the Silicon Graphics Indigo, subtle differences in color can be used to tone down dense characters such as bullets by using a gray color to even out their value in relation to less-dense letter forms in the same font.

Underline has become very common as a method of emphasis on the computer screen. It has been used, along with a second color, to mark hypertext links within text in systems such as the Help system included in Microsoft Windows. When the underline cuts through the descenders in a word, it can seriously inhibit legibility. Some screen fonts on the Macintosh leave a one-pixel gap around descenders, which improves legibility. Another interesting strategy was used first by Voyager to mark glossary links in its cd-rom products based on HyperCard and is now used by Apple for the same purpose in its online version of the Macintosh documentation in HyperCard. The method is

Figure 14
Underlining in Help for Microsoft Windows 3.1. Note how the solid underline cuts through the descenders, making it difficult to see the letter g. The dotted underline also interferes with legibility, but to a lesser degree.



to draw a two-pixel line with alternating black and white pixels, creating an optical gray. This line contrasts with the solid black of the letter form so that both the underline and the descender are clearly visible. This method is used in Voyager's Expanded Books Toolkit to mark all forms of annotation.

Shipping the Fonts with the Text

Many factors limit the designer's choice of fonts for screen design. Given all the problems listed above, choosing a screen font is more an act of triage than of aesthetic preference. There are very few fonts that present reasonable character bitmaps and intercharacter spacing. Added to these limitations is another limit: the fonts available on the reader's system.

When a print designer chooses a font for the printed page, the printing process locks that design decision into the fiber of the paper. It is the typesetting service bureau that has purchased the font from the appropriate type foundry, not the reader of the book. When we design a publication for the computer screen today we have to worry about whether our intended reader already owns the font we choose.

The current situation is intolerable and is changing fast. (See "The Role of Fonts in Electronic Delivery" on pp. 8-9 of our *Special Report from Seybold Seminars '93* for a summary of recent developments in this area.) However, these changes are being driven by the needs of an emerging "electronic delivery" industry that is premised on the need to deliver images of printed pages among computer systems. As we hope to have made clear, an application that sets a line of type for a specific column width on a "page" other than the screen itself is likely to have already compromised screen legibility.

The vendors who sell the fonts understandably do not want to give away the font information with any document that happens to use them if that font information can be used in other documents. The vendors that sell software to render bitmap fonts from any outline master want us to believe that this will be good enough. There are file size and performance tradeoffs to consider along with economic and legal issues, but it is not clear what approach will best serve the needs we are addressing here.

Screen legibility will still depend entirely on how well a font is rasterized in combination with how well character, word and line spacing are managed by an application. All the rounding up and rounding down that we currently see in the ATM and TrueType rasterizers are "good enough" today for only a small number of font designs.

Current Options

The table on page 14 summarizes the ways in which control of text is supported in a sampling of multimedia application packages. This table is certainly not meant to be an exhaustive list, but the six applications listed are representative of the kind of software tools available on the market today. As is readily apparent from the table, most of the current multimedia tools are sorely lacking in typographic control. For example, of these six, only Interleaf WorldView supports hyphenation of justified text.

Conclusion

Clearly we are still at a very early stage in the development of tools for electronic publications. It has been six years since Apple's HyperCard introduced a large audience to an entirely new concept of interactive publications. Tools for creating and delivering animation, sound and video have been improving slowly. The text universe has evolved from bitmaps for a few sizes of a few fonts to the present world of arbitrary sizes for hundreds of fonts.

When desktop publishing began in the mid-1980s, the focus was on how well a program could support the combination of text and graphics on the page. Support for typography was very basic in the first version of a seminal product such as Aldus PageMaker. The typographic and general design sophistication of many of the pioneer users was also very basic, as was often painfully clear in the newsletters and brochures they turned out with these revolutionary tools. However, once the desktop publishing products began to have major impact in the graphic arts field, the need for improved typographic support became obvious and the software vendors responded. The "serious" applications quickly differentiated themselves by offering most of the sophistication once found in the best typesetting systems. Those who did not respond were left with little or no market share.

We expect to see a similar evolution in the support for type on the computer screen among the multimedia authoring tools in the coming years. Once the amazement at seeing little movies and watching the animated diagrams wears off, we hope the market will begin seriously to ask "how legible is the text on the screen?"

How screen bitmaps of a font are rendered is only a part of that answer. The resolution of computer screens will increase. Font rasterizers will improve. Whatever the current state of the art, we can support legibility best with proper tools for the art of typography. This requires not only better screen bitmaps but also controls all of the elements we've discussed here. Typography is a whole system with its own homeostasis and balance. We cannot create the best results with tools that control only a small part of a whole system.

Much of the audience today is not sensitive to the difference between good and bad type on the screen. This will change quickly. As the amount of text we are asked to read on the screen increases, the legibility of that text will become a critical point of differentiation. Just because it is now digital information, we should not forget that the information that matters is locked in the text. The hardware and software that gives the designer tools to make text legible will win the biggest audience.

Sources and Additional Reading

Charles Bigelow and Donald Day, "Digital Typography" *Scientific American*, Vol. 249, No. 2, August 1983, pp. 106-119.

Charles Bigelow, "Font Design for Personal Workstations" *Byte,* January 1985, pp. 255-270.

Charles Bigelow and Kris Holmes, "The Design of Lucida: an Integrated Family of Type for Electronic Literacy" in *EP86: Text Processing and Document Manipulation,* J.C. van Vliet (ed.), Cambridge: Cambridge University Press, 1986, pp. 1-17.

Robert Bringhurst, The *Elements of Typographic Style,* Vancouver, BC: Hartley & Marks, 1992

Williams M. Ivins, Jr., *Art & Geometry, A Study in Space Intuitions,* New York: Dover Publications, 1964

Paul Kahn, Julie Launhardt, Krzysztof Lenk and Ronnie Peters, "Design of Hypermedia Publications: Issues and Solutions" in *EP90: Proceedings of the International Conference on Electronic Publishing, Document Manipulation & Typography,* R. Furuta (ed.), Cambridge: Cambridge University Press, 1990, pp. 107-124.

Alexander Lawson, *Anatomy of a Typeface,* Boston: Godine, 1990.

Margaret R. Lazzari and Clayton Lee, *Art and Design Fundamentals,* New York: Van Norstrand Reinhold, 1990.

John Lewis, *Typography Basic Principles: Influences and Trends Since the 19th Century,* New York: Reinhold Publishing Company, 1966.

Brody Neuenschwander, *Letterwork, Creative Letterforms in Graphic Design,* London: Phaidon Press, 1993.

Ruedi Rüegg, *Basic Typography: Design with Letters,* Zürich: ABC-Verlag, 1989.

Herbert Spencer, *The Visible Word,* Royal College of Art, London, 1968

Miles A. Tinker, *Legibility of Print,* Ames: Iowa State University Press, 1963.

Miles A. Tinker, *Basis for Effective Reading,* Minneapolis: University of Minnesota Press, 1965.

Jan Tschichold, *The Form of the Book: Essays on the Morality of Good Design,* Point Roberts, Wash.: Hartley & Marks, 1991.