Zebrackets: A Pseudo-Dynamic Contextually Adaptive Font

Michael Cohen

Abstract

A system is introduced that increases the information density of textual presentation by reconsidering text as pictures, expanding the range of written expression. Indicating nested associativity with stripes, *Zebrackets* uses small-scale horizontal striations, superimposed on parenthetical delimiters. This system is implemented as an active filter that re-presents textual information graphically, using adaptive pseudo-dynamic character generation to reflect a context that can be as wide as the document.

0 Introduction

To represent parenthetical expressions, traditionally typewritten documents use parentheses, "( )", for first-level subphrases, extended by (square) brackets, "[ ]", for doubly nested phrases (parentheses within parentheses), and alternating the two sets of delimiters for the rare more deeply nested phrases. Parentheses and brackets are overloaded; they are used in prose for delimiting subordinate expressions, appositives, citations and cross-references, and in mathematical formulae and computer programs for associative precedence, array subscripts, numeric ranges, function parameters and arguments, as well as special interpretations (like "$((n))_m$" denoting "n mod[ulo] m"). Editors also use brackets to set off editorial substitution and interpolation ("[sic]"), ellipsis ("[...]"), elision ("[expletives deleted]"), etymology ("[MF *braguette* codpiece, fr. dim. of *brague* breeches, fr. OProv *braga*, fr. L *braca*, fr. Gaulish *brāca*, of Gmc origin; akin to OHG *bruoh* breeches – more at BREECH]"), etc. For literature and journalism, these conventions have been adequate, since the reader could usually parse the subphrases. Extended schemes have used (curly or set) braces (a.k.a. "bracelets"), "{}", and angle brackets (a.k.a. "inequality signs"), "<>", to indicate more deeply nested phrases.[1] But especially for non-natural languages, in which stacks of association are not only comprehensible but necessary and encouraged, a more extensible scheme is needed.

Using size to denote nesting (or any other kind of) level doesn't work, since juxtaposed expressions, at the same parenthetical depth, might require different sized delimiters for aesthetic (lexicographical) reasons. "(A * (B + C))" suffices, but "$(\begin{pmatrix} w & x \\ y & z \end{pmatrix} * (\mathrm{B} + \mathrm{C}))$" starts to degrade. Likewise, $\overbrace{\text{over}}$ / $\underbrace{\text{under}}$ –$\overline{\underline{\text{line}}}$/ $\overbrace{\underbrace{\text{brace}}}$ schemes break down across line boundaries.

Bit-mapped terminals and high-resolution printers suggest the possibility of more elaborate presentations [Rub88] which exploit underutilized human visual acuity. Figure 1 shows some simple axes of variation under LaTeX/TeX [Lam86, Knu84]. Combining a computer filter (to analyze the text and automatically prepare appropriate semi-custom fonts) with an extra coding dimension orthogonal to standard display techniques, we can add more content to the ordinary printed display.

Zebrackets[2] [Coh92a, Coh92b] extends parentheses and square brackets by striping them systematically according to an index reflecting their order in the text. Each index of the respective delimiter pairs is cast into a binary pattern, which is superimposed on the characters as striations. The striping is adaptively chosen, so that the complexity of the parenthesized expression determines the spacing of the striations. Informal experiments suggest that users tend to look only for matching delimiters, but alternate encoding schemes are also possible. Some of these special-purpose modes are outlined later in the Discussion.

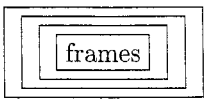
1 Examples

1.1 Chemical Compound

Figure 2 shows the the application of *Zebrackets* to the chemical formula for a popular roach control system, as shown on its box. What is the "matching bookend" to the parenthesis before "3-" in the second line? A quick scan of the zebracketed version finds the parenthesis closing the "(3-" association.

[1] Note that the open and close quotation marks are themselves, like yin and yang, pairwise delimiters.

[2] This paper uses the following orthographic conventions: the name of the described system is italicized; computer commands are in typewriter style; and most everything else (except for the bibliography, which has its own conventions) is in roman type style. Therefore, "*Zebrackets*" refers to the system described here meant to elegantly display nested associations, "`zebrackets`" to the eponymous computer filter for invoking this system, and "zebracketed" to the corresponding effect.

- ([{<delimiter shape>}])
- indentation
  - outlines
- 

- (((((((parenthesis (or type)) size)))))))
- color, greyscale, or shading
- dynamic effects
  - momentary highlight (during editing)
  - synchronous flashing
- (explicitly) tagged delimiters
- overlines and underlines
- (superscripted) overbraces and/or (subscripted) underbraces

Figure 1: Some traditional and non-traditional ways of typographically indicating nested associativity

1.2 LISP

The LISP programming language relies on parentheses for delimiting lists,[3] the language's basic data structure. This LISP function performs a generalized "inclusive or":

```
(DEFUN ANY (LST)
 (COND ((NULL LST) NIL)
       ((CAR LST) T)
       (T (ANY (CDR LST))) ) )
```

Here is the same code, but written with zebrackets, which elucidates the associations:

```
(DEFUN ANY (LST)
 (COND {(NULL LST) NIL}
       {(CAR LST) T}
       {T (ANY (CDR LST))} ) )
```

[3] The name "LISP" is acronymic for "**list processing language**", but has been jokingly expanded as "**l**ots of **i**ncessantly **s**illy **p**arentheses".

1.3 Objective-C

Figure 3 shows a line from an Objective-C program, with and without *Zebrackets*.

1.4 Logic

Figure 4 shows the first order predicate calculus notation indicating that symmetry and transitivity imply reflexivity. *Zebrackets* illuminates the precedence.

1.5 Association Beyond Single Parenthetical Pairs

Just as pairs of identically valued parentheses point at each other, like matching bookends, so do multiple sets of same-valued delimiters associate beyond the scope of a single pair of parentheses. Here the zebrackets are not just reinforcing patterns already present, but are adding new information, distributed across the text:

> An angle bracket ">" {"<"} may be used to write {read} Unix file{s} to {from} standard output {input}.

ACTIVE INGREDIENT: Hydramethylnon [tetrahydro-5, 5-dimethyl-2(1*H*)-pyrimidinone(3-[4-(trifluoromethyl)phenyl]-1-(2-[4-(trifluoromethyl) phenyl]ethenyl)-2-propenylidene)hydrazone]

ACTIVE INGREDIENT: Hydramethylnon [tetrahydro-5, 5-dimethyl-2(1*H*)-pyrimidinone(3-[4-(trifluoromethyl)phenyl]-1-(2-[4-(trifluoromethyl) phenyl]ethenyl)-2-propenylidene)hydrazone]

Figure 2: Chemical Compound, before and after *Zebrackets*

```
[inspectorPanel setAccessoryView:[[[[accessory contentView] subviews]
                                   objectAt:0] removeFromSuperview]];

[inspectorPanel setAccessoryView:[[[[accessory contentView] subviews]
                                   objectAt:0] removeFromSuperview]];
```

Figure 3: Objective-C code, before and after *Zebrackets*

$$\forall P\ \forall v\ \forall w\ \forall x\ \forall y\ \forall z\ ((P(v,w) \Rightarrow P(w,v)) \wedge ((P(x,y) \wedge P(y,z)) \Rightarrow P(x,z)) \Rightarrow P(u,u))$$

$$\forall P\ \forall v\ \forall w\ \forall x\ \forall y\ \forall z\ ((P(v,w) \Rightarrow P(w,v)) \wedge ((P(x,y) \wedge P(y,z)) \Rightarrow P(x,z)) \Rightarrow P(u,u))$$

Figure 4: First order predicate calculus, before and after *Zebrackets*

and

$$(\ln|(\ln|(x^2-2)|)|)' = (\ln|(x^2-2)|)^{-1}\ (x^2-2)^{-1}\ (x^2-2)'$$

The visual coupling of related expressions is also useful when parenthetical expressions cross, violating LIFO (last in ⇒ first out) protocol and subverting the "most recent unmatched" association. One domain in which this happens is when part of a character string's postfix is the same as another's prefix. (This recalls the [American TV show] Wheel of Fortune's "Before and After" category, or Emacs' "Dissociated Press" function [Sta86].)

⊙ associativity: A ⊙ B ⊙ C = (A ⊙ (B) ⊙ C)

(reference (manual) labor)

(Ze(bra)ckets)

(FM/(AM)/PM)

2 Implementation

Zebrackets glyphs are implemented as extensions of fonts in the Computer Modern typeface. The implementation of *Zebrackets* comprises two aspects: a filter to generate permuted invocations of the underlying parentheses and brackets, and the delimiter fonts themselves. The two-pass filter parses selected text. The first pass establishes the maximum number of stripes needed, and generates the necessary METAFONT [Knu86] files. (For the simplest mode, in which each pair of delimiters is assigned a unique index, the maximum number of stripes is the number of bits needed to represent its highest index, $\lceil \lg_2 |\text{delimiter pairs}| \rceil$.) Using the context established by the first pass, the second pass replaces each parenthesis or bracket with the LaTeX code invoking its respective zebracketed version.

For example, running the `zebrackets` filter on "`(a * (b + c))`" determines that only one potential stripe is needed, replaces the source text with "`{\pmcone \symbol{0}}a * {\pmcone \symbol{1}}b + c{\pmcone \symbol{3}}{\pmcone \symbol{2}}`", and generates the `pmcone12.mf` file, which together yield "(a * (b + c))" at image time.

By having indirected the glyphs one extra level, *Zebrackets* implements a meta-METAFONT. Dynamic fonts exploit what is sometimes called "dynamic programming", which is basically lazy evaluation of a potentially sparse domain. *Zebrackets* is implemented as a precompiler, like a macro processor, that replaces vanilla parentheses with zebracketed and emits METAFONT source that will be invoked at preview (`TeXview` on the NeXT) or printing (`dvips`) time. Since each character is determined at edit-time, this implementation is context-sensitive and adaptive, but not purely dynamic [AB89]. Because the locality disambiguates *Zebrackets*' indices, the scope of a zebracketed expression is typically a

single formula or expression. Further, since each LaTeX `\newfont` must be declared at most once, it is easiest to simply include a list of all the potential *Zebrackets* fonts at the top of each file. Therefore I have termed *Zebrackets* 'pseudo-dynamic', since the automatic adaptive character specification can be conceptually lumped together with the compilation (via `latex`) and imaging, but the actual specification usually involves human-specified ranges for `zebracketsing`.

Zebrackets adopts a minimalist "less (ink) is more (data)" philosophy [Tuf83], adding information to regular parentheses and brackets by resetting some pixels in the characters. Because of the tendency of white features to bleed out into a black background, the striations need not be terribly thick to be perceivable. In the examples shown, the stripes are 1 pt. ($\frac{1}{72.27}$ inches) thick, subsuming about 2.6 arc-minutes (≈ 0.044 degrees ≈ 0.77 milliradians) of visual angle (assuming a reading distance of 18 inches). This is greater than the accepted (if heuristic) industrial minimum for visual acuity, 1 arc-minute.

Since there are usually more ascending than descending characters, readers (of English) tend to look slightly above the line of print, deriving meaning from the "top coastline" (upper half) of the text. Therefore, the index of the parenthetical pair is encoded with the LSB (least significant **bit**) at the top of the parenthesis or bracket. *Zebrackets* are generated singly, although they occur pairwise in well-formed expressions. (When are parenthetical delimiters unmatched? In expressions like 'electroglyphs' or ASCII 'smilies' [Tem89] that suggest a rotated human face `:-)`.) For both square and curved delimiters, the stripes are drawn horizontally (rather than radially), so that the reader might imagine an invisible line drawn through the intermediate text. Since the delimiter pairs are symmetrical [Lan92], they can be conceptually associated, and since the bands are aligning, they can also be visually associated, like toothpicks holding bread around a thick sandwich.

3 Discussion

For single levels of nesting, the extended parentheses and brackets are identical to the unenhanced, since an index of zero leaves the delimiters unbanded in *Zebrackets*' default positively-coded scheme, chosen to preserve the backwards compatibility of the curve substrate.

For deeper levels, *Zebrackets* tries to maintain this backwards compatibility by being non-distracting to users who don't know about it. It is, by design, "just noticeable", right over the limen, or edge of perceptibility. The added information is meant to be clear to the user actively seeking it, and transparent to any user not actively seeking it. (Such an effect is like making something a little smaller to call attention to it.) By designing a scheme that is both noticeable and ignorable, one is obtained that is unambiguous but unobtrusive, unmistakable but unassuming.

In practice, however, *Zebrackets* has some problems: For linear reading without searching, zebrackets can actually be distracting, introducing high-frequency noise (that looks like printer spotting). On the other hand, zebracket striations can be difficult to see, especially for readers with less sharp eyes, and zebracketed documents are not robust under (perhaps repeated) copying by low-resolution devices (like most faxes). The useful limit of the current system seems to be around six (allowing $2^6 = 64$ different versions of each pair of delimiters). For overly rich expressions that exceed visual acuity, *Zebrackets* can be limited to a fixed stripe depth, wrapping around (repeating) the indexing scheme if the delimiters exhaust the range of uniquely encodable depths.

Variations that overcome these limitations are possible. For instance, grayscale striations (not yet implemented) might disappear at normal reading speed, but be visible when doing a detailed search. Alternatively, using black stripes to tick the parentheses, instead of dropping out (white) segments, is more legible, if less inconspicuous:

```
(DEFUN ANY (LST)
 (COND ((NULL LST) NIL)
       ((CAR LST) T)
       (T (ANY (CDR LST))) ) )
```

Further, invocation of different encoding modes and graphical rhythms, perhaps for special purposes, is straight-forward. Indexing the streaks "inside-out", (so that the outer delimiters pairs have a higher index (than inner)), orders the evaluation of expression trees. Displaying breadth (or depth (or both)), instead of an incremental index, is useful for visualizing expression complexity. Striping the delimiters "upside-down", from the bottom, might be better for languages (like Hebrew) that carry more information in the "lower coastline".

4 Conclusion

These utilities recall the counter-intuitive advice "To clarify, add detail" [Tuf90]. *Zebrackets* is a tool

in a suite of prettyprinters that start to treat characters as pictures, with attendant increases in information/ink value: denser data without loss of legibility, intensifying without interfering.

Documents should look like what they mean: context after content, form after function, process after product, and style after substance. Creative orthography frees words from traditional (technologically imposed) constraints, allowing textual representation of multidimensional concepts by projecting multilayered structure into linear text. Extended electronic typography provides additional parsing cues and differentiates between heretofore duplicate symbols, stretching existent presentation styles without breaking into new modalities.

The handwritten "publishing" of pre-Gutenberg scribes was arbitrarily subtle, with its attendant human caprice (and mistakes). Printing can be thought of as having rigidified information transmission. The research described here loosens some of that determinism, not by randomizing the presented information, but by softening the digitized boundaries, thereby expanding the range of expression.

The notion of a fixed alphabet font is inherently limited, even one extended into a family by techniques like weighting, italicization, emboldening, and local contextual tools like ligature and kerning. Computers offer the potential of arbitrarily adaptable glyphs, permuted in subtle if significant ways, depending on their global context, to heighten legibility (readability, balance, or proportion) or evoke emotions that reinforce or complement the words and ideas.

Zebrackets' integral characterization of the text yields a discrete specification of a font; real numbers would allow for continuous variation, expanding even further the ability to custom-tailor a font for a context. Such variety might manifest as arbitrarily soft typefaces, perhaps employing greyscale or dynamic effects, or tuned by the user, to match visual acuity. Contextual fonts like *Zebrackets* indicate evolving modes of written representation, algorithmic descriptions driving adaptive displays, as style catches up to technology.

◇ Michael Cohen
The University of Aizu
Aza Kami-iawase 90, Oaza
Tsuruga
Ikki-machi, Aizu-Wakamatsu-shi
Fukushima-ken 965
Japan
mcohen@u-aizu.ac.jp

References

[AB89] Jacques André and Bruno Borghi. Dynamic fonts. In *Proc. Int. Conf. on Raster Imaging and Digital Typography*, pages 198–204, Lausanne, Switzerland, October 1989.

[Coh92a] Michael Cohen. Blush and Zebrackets: Large- and Small-Scale Typographical Representation of Nested Associativity. In *Proc.* IEEE *Workshop on Visual Languages*, pages 264–266, Seattle, September 1992.

[Coh92b] Michael Cohen. Large- and Small-Scale Typographical Representation of Nested Associativity. *Visible Language*, 23(3/4), Summer/Autumn 1992.

[Knu84] Donald E. Knuth. *The TEXbook.* Addison-Wesley, 1984. 0-201-13448-9.

[Knu86] Donald E. Knuth. *The METAFONTbook.* Addison-Wesley, 1986. 0-201-13444-6.

[Lam86] Leslie Lamport. *LATEX: A Document Preparation System.* Addison-Wesley, 1986. 0-201-15790-X.

[Lan92] John Langdon. *Wordplay: ambigrams & reflections on the art of ambigrams.* Harcourt Brace Jovanovich, 1992. 0-15-198454-9.

[Rub88] Richard Rubinstein. *Digital Typography: An Introduction to Type and Composition for Computer System Design.* Addison-Wesley, 1988. 0-201-17633-5.

[Sta86] Richard M. Stallman. *GNU Emacs Manual.* Free Software Foundation, 1000 Mass. Av.; Cambridge, MA 02138, fifth edition, December 1986.

[Tem89] Brad Templeton, editor. `Rec.Humor.Funny/TeleJoke` *Computer Network Humour Annual.* ClareNet Communications Corp., 1989.

[Tuf83] Edward R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, 1983.

[Tuf90] Edward R. Tufte. *Envisioning Information.* Graphics Press, 1990.