

ScatterType: a Reading CAPTCHA Resistant to Segmentation Attack

Henry S. Baird and Terry Riopka

Computer Science & Engineering Dept
Lehigh University
19 Memorial Dr West
Bethlehem, PA 18017 USA

E-mail: {baird|riopka}@cse.lehigh.edu

URL: www.cse.lehigh.edu/~baird

ABSTRACT

A reading-based CAPTCHA, called ‘ScatterType,’ designed to resist character-segmentation attacks, is described. Its challenges are pseudorandomly synthesized images of text strings rendered in machine-print typefaces: within each image, characters are fragmented using horizontal and vertical cuts, and the fragments are scattered by vertical and horizontal displacements. This scattering is designed to defeat all methods known to us for automatic segmentation into characters. As in the BaffleText CAPTCHA, English-like but unspellable text-strings are used to defend against known-dictionary attacks. In contrast to the PessimPrint and BaffleText CAPTCHAs (and others), no physics-based image degradations, occlusions, or extraneous patterns are employed. We report preliminary results from a human legibility trial with 57 volunteers that yielded 4275 CAPTCHA challenges and responses. ScatterType human legibility remains remarkably high even on extremely degraded cases. We speculate that this is due to Gestalt perception abilities assisted by style-specific (here, typeface-specific) consistency among primitive shape features of character fragments. Although recent efforts to automate style-consistent perceptual skills have reported progress, the best known methods do not yet pose a threat to ScatterType. The experimental data also show that subjective rating of difficulty is strongly (and usefully) correlated with illegibility. In addition, we present early insights emerging from these data as we explore the ScatterType design space — choice of typefaces, ‘words’, cut positioning, and displacements — with the goal of locating regimes in which ScatterType challenges remain comfortably legible to almost all people but strongly resist machine-vision methods for automatic segmentation into characters.

*

Keywords: CAPTCHAs, human interactive proofs, document image analysis, abuse of web sites and services, human/machine discrimination, Turing tests, OCR performance evaluation, document image degradations, legibility of text, segmentation, fragmentation, Gestalt perception, style-consistent recognition

1. INTRODUCTION

In 1997 Andrei Broder and his colleagues at the DEC Systems Research Center, developed a scheme to block the abusive automatic submission of URLs to the AltaVista web-site [Bro01,LBBB01]. Their approach was to challenge a potential user to read an image of printed text formed specially so that machine vision (OCR) systems could not read it but humans still could. Since that time, inspired also by Alan Turing’s 1950 proposal of methods for validating claims of artificial intelligence [Tur50], many such CAPTCHAs — Completely Automated Public Turing tests to tell Computers and Humans Apart — have been developed, including CMU’s EZ-Gimpy [BAL00, HB01], PARC’s PessimPrint [CBF01] and BaffleText [CB03], Paypal’s CAPTCHA (www.paypal.com), and Microsoft’s CAPTCHA [SSB03]. In addition to these, which have been describe in the literature, many others have been put into practice. Examples of these are shown and critiqued in Figures 1–4.

*Accepted for publication in *Proceedings, IS&T/SPIE Document Recognition & Retrieval XII Conference*, San Jose, CA, January 16–20, 2005.



Figure 1. Example of an AltaVista challenge: characters are chosen at random, then each is assigned to a typeface at random, then each character is rotated and scaled, and finally (optionally, not shown here) background clutter is added. The fact that the characters are spaced widely apart invites segment-then-recognize attacks.



Figure 2. Example of a simplified Yahoo! challenge (CMU’s “EZ GIMPY”): an English word is selected at random, then the word (as a whole) is typeset using a typeface chosen at random, and finally the the word image is altered randomly by a variety of means including image degradations, scoring with white lines (shown here), and non-linear deformations. The use of a single known typeface makes this easy to segment into characters.

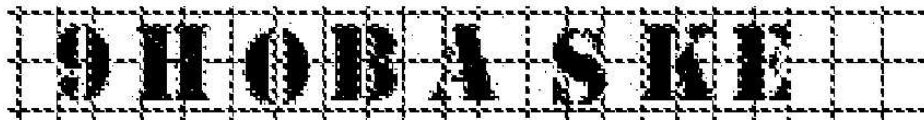


Figure 3. Example of a PayPal challenge: alphabetic characters and numerals are chosen at random and then typeset, spaced widely apart, and finally a grid of dashed lines is overprinted. The wide character spacing invites segmentation attacks.



Figure 4. Example of a PessimialPrint challenge: an English word is chosen at random, then the word (as a whole) is typeset using a randomly chosen typeface, and finally the word-image is degraded according to randomly selected parameters (within certain ranges) of the image degradation model. Judicious choice of image restoration preprocessing can make character-segmentation straightforward.

Fully or partially successful attacks on some of these CAPTCHAs have been reported. EZ-Gimpy has been broken by a lexicon-driven and word-shape recognition attack [MM03]. PessimPrint has been shown to be vulnerable to image restoration methods followed by conventional segment-then-recognize OCR [CB03]. We believe that many, perhaps most, CAPTCHAs now in use are vulnerable to (possibly custom-tailored) preprocessing that segments the words into characters, followed by off-the-shelf or slightly customized OCR.

CAPTCHAs that appear to us to be unusually resistant to segmentation attacks include BaffleText and the Microsoft CAPTCHA.

2. MOTIVATION

These observations motivated us to investigate CAPTCHAs which resist character-segmentation attacks. Our approach is to fragment each character image using horizontal and vertical cuts, then allow the fragments to drift apart to the point that it is no longer straightforward automatically to reassemble them into characters. We have observed that, despite severe scattering, human reading skill does not deteriorate rapidly (more on this later). This occurs, we suspect, largely because we do not apply image degradations such as blurring, thinning, and additive noise (cf. [Bai02]) and so we do not obscure style-specific shape minutiae in the fragments. Thus the character fragments retain many recognizable primitive shape properties — including stroke width, serif form, curve shape, etc — which encode typeface-specific “style” information which then, in turn, assists Gestalt perception integration at the word level.

Style-conscious pattern recognition methods — which automatically exploit knowledge that an image was generated in a single style (e.g. one typeface, one person’s handwriting, a certain level of image quality), but without knowledge of *which* style — have been shown to lower error rates [SN05,VN05]. The study of these methods is still in its early stages. While significant progress is being made, we judge that it is unlikely that an effective attack on ScatterType is feasible using today’s best-understood methods. In our view, a CAPTCHA which depends on style-conscious recognition for success is likely to resist automatic attack for many years.

3. SYNTHESIZING SCATTERTYPE CHALLENGES

ScatterType challenges were synthesized by pseudorandomly choosing: (a) a text-string; (b) a typeface; and (c) cutting and scattering parameters.

The text strings were generated using the pseudorandom variable-length character n -gram Markov model described in [CB03], and filtered using an English spelling list to eliminate all but a few English words. As in [CB03], this is intended to protect against lexicon constrained recognition attacks. Random strings were not used because psychophysical evidence suggests that familiarity — even at the low level of frequently occurring short strings of characters — improves human legibility. The BaffleText trial also indicated that the use of English-like “words” raised the comfort level of subjects: an important feature since many people feel irritated or threatened by CAPTCHAs. In the trials, no word was ever used twice — even with different subjects — to ensure that mere familiarity with the words would not affect legibility.

The typefaces used were twenty-one FreeType fonts listed in Figure 14.

Cutting and scattering are applied, separately to each character (more precisely, to each character’s image within its own ‘bounding box’); then the modified character images are combined into a text-string image. A scaling dimension (the “base length”) equal to the height of the shortest character in the alphabet (in our case, lowercase ‘o’) is used to achieve comparable results across different text sizes. The following image operations are performed pseudorandomly to each character separately, controlled by the following parameters.

Cutting Fraction Each character’s bounding box image is cut into rectangular blocks of size equal to this fraction of the base length. The cutting fraction, in general, can be different in the x and y directions: in the trial we report, they were set equal. The resulting x & y cut fractions are held constant across all characters in a text string, but the offset locations of the cuts are chosen randomly uniformly independently for each character.

ScatterType Parameter	Range used in Trial
Cut Fraction (both x & y)	0.25-0.40
Expansion Fraction (both x & y)	0.10-0.30
Horizontal Scatter Mean	0.0-0.40
Vertical Scatter Mean	0.0-0.20
Scatter Standard Error (both h & v)	0.50
Character Separation	0.0-0.15

Figure 5. ScatterType parameter ranges selected for the human legibility trial.

Expansion Fraction Fragments are moved apart by this fraction of base length. The expansion fraction, in general, can be different in x and y: in this trial, they were set equal. The resulting fractions are held constant across all characters in a text string.

Horizontal Scatter Each row of fragments (resulting from horizontal cutting) is moved horizontally by a displacement chosen independently for each row: this displacement is a positive random number, expressed as a fraction of the base length, and distributed normally with a given mean and standard error. Adjacent rows alternate left and right movements.

Vertical Scatter Each fragment within a row (resulting from vertical cutting) is moved vertically by a displacement chosen randomly independently for each fragment: this displacement is a positive random number, a fraction of the base length, distributed normally with a given mean and standard error. Adjacent fragments within a row alternate up and down movements.

Finally, once every character image has been cut and scattered, the resulting images are reintegrated (by pixel-wise Boolean OR) into the final text-string image, governed by this final parameter:

Character Separation The images of cut-and-scattered characters are combined (by pixel-wise Boolean OR) into a final text string image by locating them using the original vertical coordinate of the bounding box center, but separating the boxes horizontally by this fraction of the width of the narrower of the two adjacent characters’ bounding boxes. Character separation may be positive or negative: negative values allow character images to overlap.

Note that, whereas horizontal scatter is constrained in that all fragments within a row are displaced the same amount in the same direction, vertical scatter is not constrained: that is, the fragments within a column move up and down independently. Scattering was implemented in this direction-sensitive manner in order, on the one hand, to make it difficult for “fragment stacking” methods to reassemble each character, and, on the other hand, to provide enough proximity to assist human perception of vertical grouping among fragments in different cut rows.

Character Separation was designed to be a fraction of the width of the narrower of each pair of adjacent characters (instead of a fraction of the base length) in order to prevent thin characters from being “swallowed up” by wider adjacent characters.

Before the human legibility trial (described in the next Section), we ran small-scale pilot experiments to help us select ranges for ScatterType parameters that would be likely to yield roughly equal numbers of legible and illegible images: the parameter ranges we selected are given in Figure 5. To generate a ScatterType challenge, a full set of nine parameters were chosen randomly uniformly within each range independently, then applied to generate the challenge image. As we will see, these yielded roughly half-and-half legible and illegible images: a rich space for data analysis. These challenges are illustrated, for three levels of difficulty, in Figures 7–9.

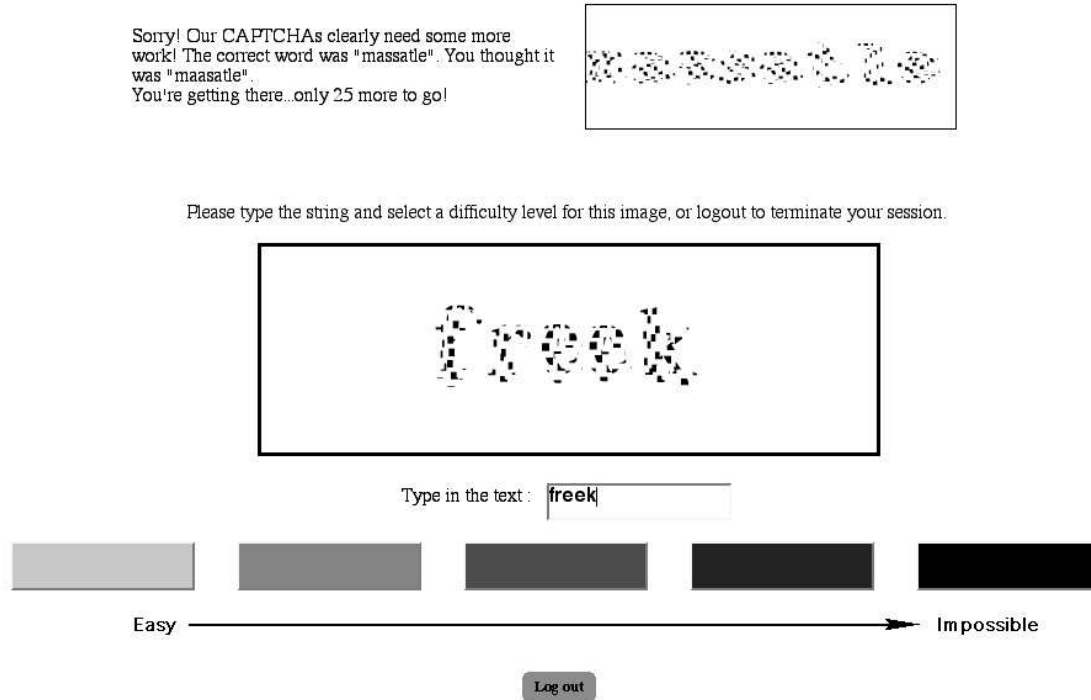


Figure 6. An example of a ScatterType legibility trial challenge page. The Difficulty Level radio buttons (marked 'Easy' to 'Impossible') were colored Blue, Green, Yellow, Orange, and Red. The text at the top of the page refers to the previously answered challenge.

4. LEGIBILITY TRIAL

Students, faculty, and staff in the Lehigh CSE Dept, and researchers at Avaya Labs Research, were invited to attempt to read ScatterType challenges using ordinary browsers, served by a PHP GUI backed by a MySQL database. A snapshot of the challenge page is shown in Figure6.

The text images for the challenges were displayed approximately 0.7 inches high on the monitor in an effort to maximize legibility. The psychophysical literature [LPRS85] reports that humans can read best when the subtended angle from the eye to the character height is $0.3\text{-}2.0^\circ$. Assuming the distance from the eye to the monitor screen is about 20 inches, the optimal range of character height ranges from 0.2 to 0.7 inches.

Each subject was asked, of course, to read the text and type it in, to the best of his/her ability. Then the subject indicated the perceived "difficulty level" of reading that challenge by clicking on one of five radio buttons arranged in a horizontal line, with the leftmost labeled "Easy" and the rightmost "Impossible". An arrow stretched from "Easy" to "Impossible," in order to suggest that the buttons represented a continuum of difficulty. This suggestion was reinforced by color: the buttons were colored Blue (for "Easy"), then Green, then Yellow, then Orange, and finally Red (for "Impossible"). No other instructions were given explaining how to select a Difficulty Level. No subject asked us any questions about how to assign these ratings.

Clicking on a Difficulty Level radio button advanced to the next challenge on a new page: at the top of this page, the subject was told whether or not he/she had read the previous challenge correctly and if not, was told the correct string. Thus the subjects understood how well they were doing and had many opportunities to learn and adapt to ScatterType. Some subjects seemed to us to have improved significantly by the end of 100 challenges. Each challenge was associated with the user-id of the subject, and time-stamped, so it is possible in principle to study this question quantitatively in future work.

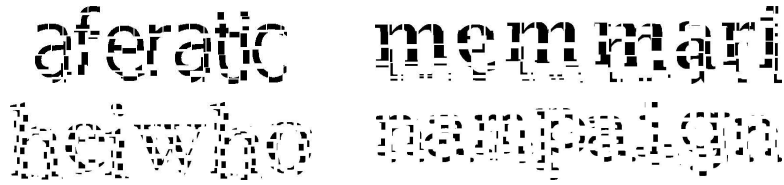


Figure 7. ScatterType challenges rated by subjects as “Easy” (difficulty level 1 out of 5). All of these examples were read correctly: “aferatic,” “memari,” “heiwho,” “nampaign.”



Figure 8. ScatterType challenges rated by subjects as being of medium difficulty (difficulty level 3 out of 5). Only one of these examples was read correctly (correct/attempt): “overch”/”overch”, “wouwould”, “adager”/”atlager”, “we-land”/”wejund”.

These challenges, by careful design, ranged from trivially easy to read to impossible to read. In order to relieve stress on the subjects, we printed cheerful encouraging remarks after each CAPTCHA, pointing out that the immature state of the CAPTCHA was principally responsible for any illegibility.

5. EXPERIMENTAL RESULTS

A total of 4275 ScatterType challenges were used in the human legibility trial: they are illustrated in Figures 7-9, at three subjective levels of difficulty: “Easy,” medium difficulty, and “Impossible.”

Human legibility — percentage of challenges correctly read — is summarized in Figure 10. Overall, human legibility averaged 53%, and exceeded 73% for the two easiest levels. Legibility was strongly correlated with subjective difficulty level, falling off monotonically with increasing subjective difficulty.

ScatterType affects the legibility of characters highly nonuniformly, as shown in Figure 11. Also, some confusions between pairs of characters are strongly asymmetric, as shown in Figure 12: note, for example, that ‘c’ was mistaken 25% of the time for ‘e’ and 14% for ‘o,’ while ‘e’ and ‘o’ were rarely mistaken for ‘c.’ Asymmetric confusions are commonplace in pattern recognition, of course, but the effect seems to have been amplified in some cases by cutting and scattering. These data suggested that by judicious pruning of characters from the alphabet we use to generate text strings, we could improve legibility. This is borne out in Figure 13: all five

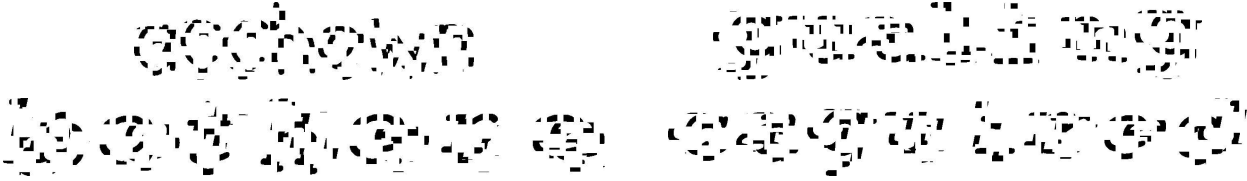


Figure 9. ScatterType challenges rated by subjects as “Impossible” (difficulty level 5 out of 5). None of these examples were read correctly (correct/attempt): “echaeva”/”acchown”, “gealthas”/”gualing”, “beadave”/”bothere”, “engaberse”/”caquired”

	Difficulty Level					
	ALL	1	2	3	4	5
Total number of challenges	4275	610	1056	1105	962	542
Percent correctly answered	52.6	81.3	73.5	56.0	32.8	7.7

Figure 10. Human reading performance as a function of the difficulty level that the subject selected.

Character	q	c	i	o	u	z	j	h	f	n	l	v	t
Confusability	2.27	1.13	0.98	0.86	0.84	0.83	0.80	0.67	0.65	0.62	0.61	0.42	0.41

Figure 11. The confusability of characters (expressed as the ratio of mistaken to correct readings). Thirteen characters are shown in descending order of confusability. A few characters were confused far more often than the rest, apparently mostly as a result of image degradations introduced by ScatterType.

difficulty levels improve nearly monotonically as confusion-prone characters are pruned, with especially strong improvement in the two easiest levels.

An analogous effect was observed with typefaces: certain typefaces preserve legibility under ScatterType far better than others (Figure 14). As with characters, judicious pruning of fonts significantly improves legibility, as shown in Figure 15: again, all five difficulty levels improve almost monotonically as the most confusion-prone fonts are pruned, with especially strong improvement in the medium-difficulty levels (levels 2 and 3).

6. DISCUSSION AND FUTURE WORK

The strong correlation between legibility and subjective difficulty level is similar to the behavior of BaffleText, where the first author showed that it could be used to engineer challenges that reliably did not irritate or anger

	b	c	e	g	h	i	l	o	q
b	0	0	1	0	3	1	1	1	0
c	0	0	25	1	0	0	0	14	0
e	0	1	0	0	0	0	0	3	0
g	0	0	5	0	0	0	0	1	1
h	21	0	0	0	0	0	1	0	0
i	1	0	0	0	0	0	21	0	0
l	1	0	0	1	2	12	0	0	0
o	0	2	30	1	0	1	0	0	0
q	0	2	3	36	2	0	0	3	0

Figure 12. A subset of the human legibility character confusion matrix. The true character classes are indicated on the Y-axis (at the left) and the interpretations of human readers on the X-axis (along the top). Shading marks character-pair confusions which are highly asymmetric.

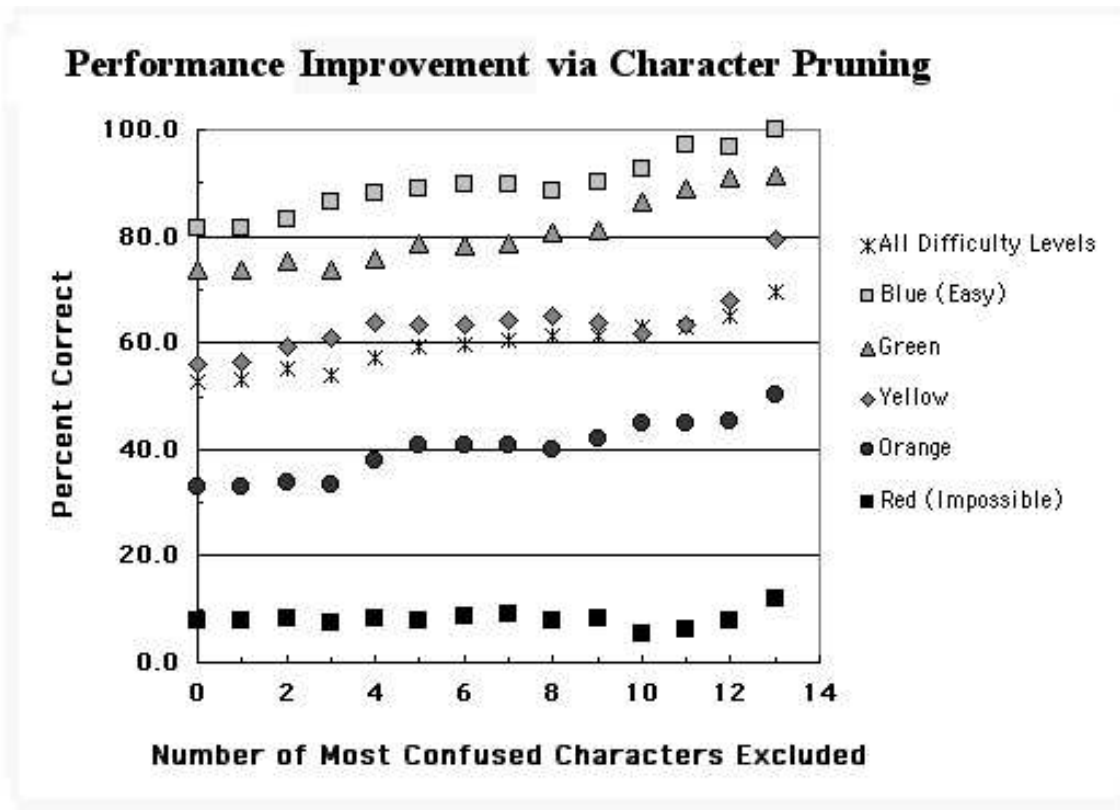


Figure 13. Improvement in legibility resulting from omitting text-strings from the ScatterType lexicon that include the most confusable characters, as a function of the number of characters pruned.

human subjects.

Our early preliminary analysis of legibility as a function of typeface and alphabet suggests that we may by judicious choices, raise legibility on the two easiest difficulty levels to above 90%.

The legibility trial data may shed light on several technically interesting and potentially practically important questions.

- What improvements in legibility that can be expected from judicious choices of generating parameters (typefaces, characters, and distributions that control cutting and scattering), so that we can control the fraction of legible challenges?
- How well can we construct classifiers for legibility, in the feature space defined by the generating parameters, so that we can automatically select legible challenges from among those that are generated?
- How well can we construct classifiers for legibility, in spaces determined by features that can be extracted from the images of the challenges *after* they are generated? An example would be the 'Image Complexity' metric that was correlated positively with legibility in the BaffleText trial.
- Given a set of ScatterType challenges, how well can we automatically select those that are likely to possess a given subjective difficulty level?

The fact that ScatterType amplifies certain character-pair confusions and not others in an idiosyncratic way might be exploitable. If further study reveals that the distribution of mistakes differ between human readers

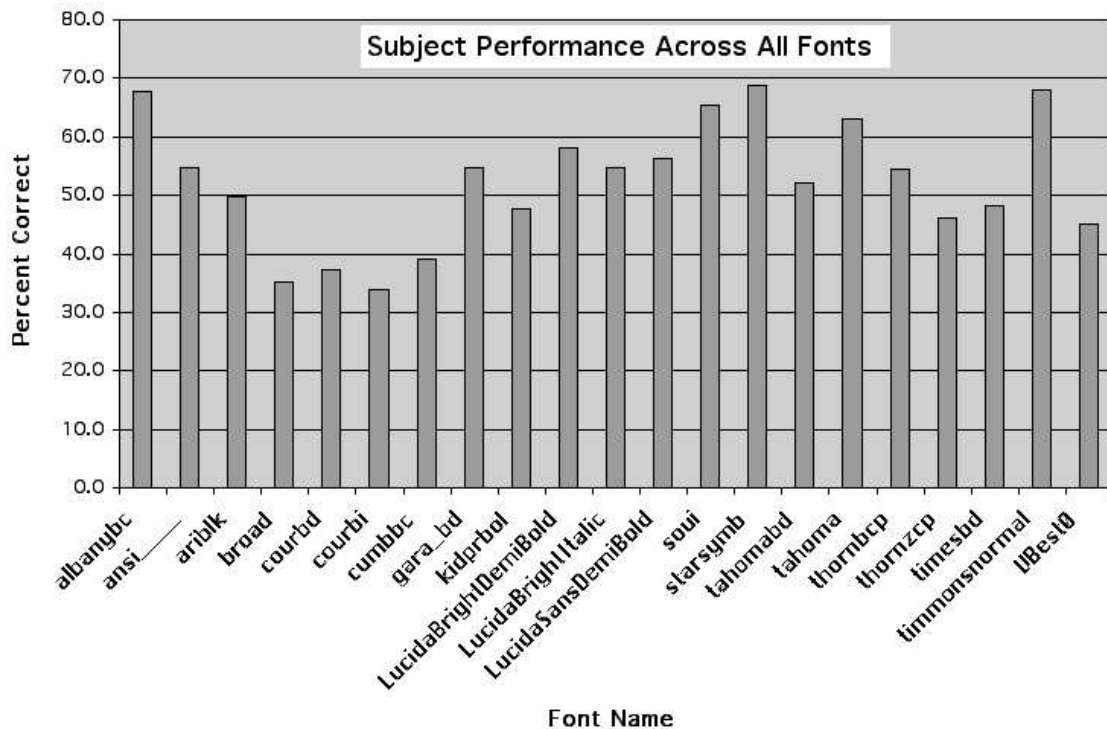


Figure 14. Legibility as a function of typeface. Some typefaces are associated with more than twice the illegibility rate as are others.

and machine vision systems, we may be able to craft policies that forgive the mistakes that humans are prone to while red-flagging machine mistakes.

Of course every CAPTCHA including ScatterType should be tested systematically using the best available OCR engines, and should be offered to the research community for attack by experimental machine vision methods. Our personal knowledge of the segment-and-recognize capabilities of commercial OCR machines — as attested by hundreds of failure cases discussed in [RNN99] — gives us confidence that they pose no threat ScatterType today or for the foreseeable future.

7. ACKNOWLEDGMENTS

We are grateful for privately communicated observations by Patrice Simard and for illuminating suggestions by Dan Lopresti, Jarret Raim, and Jon Bentley. Also, our experiments have benefited from systems software assistance by Jarret Raim, Bryan Hodgson, and David Morissette. Above all, we are thankful for the generously volunteered time of nearly 60 people — most of them students, faculty and staff in the Computer Science & Engineering Dept of Lehigh University, and some from Avaya Labs Research — who participated in the legibility trial. Warm thanks also to Ruth Tallman of the Lehigh Univ. Office of Research and Sponsored Programs, for her constructive response to our request for human subjects experiment approval.

References

- [BAL00] M. Blum, L. A. von Ahn, and J. Langford, *The CAPTCHA Project*, “Completely Automatic Public Turing Test to tell Computers and Humans Apart,” www.captcha.net, Dept. of Computer Science, Carnegie-Mellon Univ., and personal communications, November, 2000.

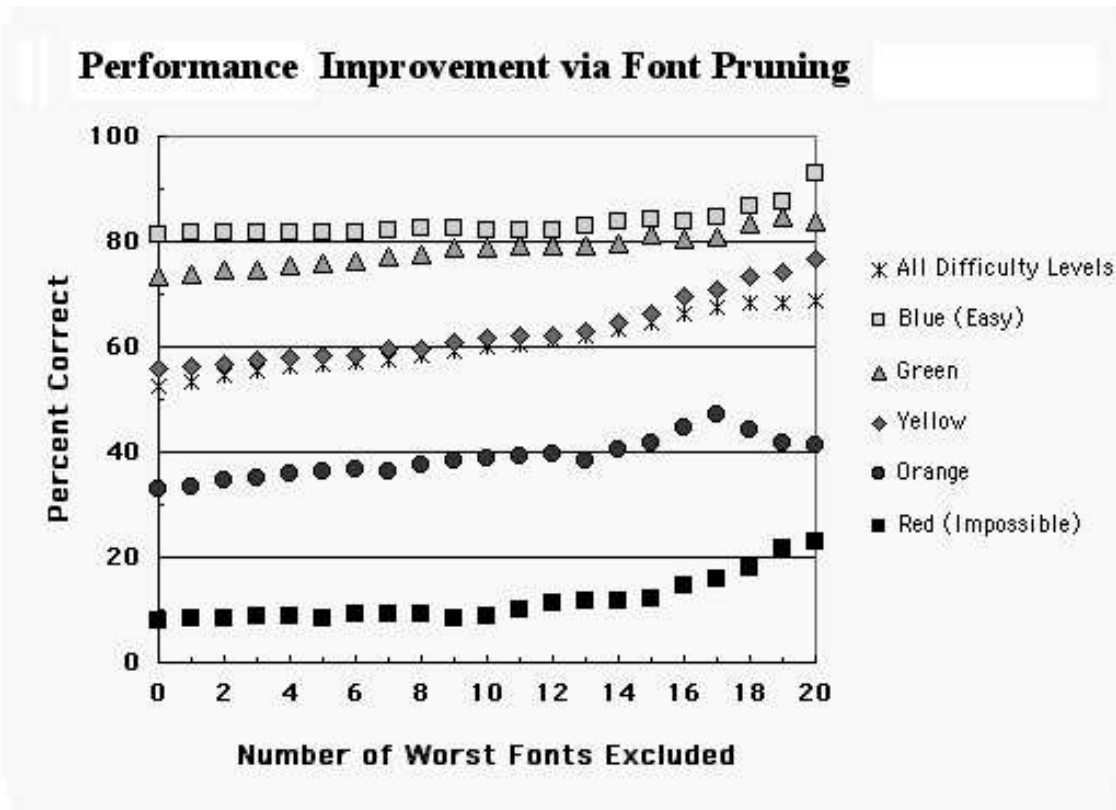


Figure 15. Improvement in legibility resulting from omitting typefaces from the ScatterType design space, as a function of the number of typefaces omitted.

- [BK02] H. S. Baird and K. Popat, "Human Interactive Proofs and Document Image Analysis," *Proc., 5th IAPR Int'l Workshop on Document Analysis Systems*, Princeton, NJ, Springer-Verlag (Berlin) LNCS 2423, pp. 507-518, August 2002.
- [Bro01] AltaVista's "Add-URL" site: altavista.com/sites/addurl/newurl, protected by the earliest known CAPTCHA.
- [CB03] M. Chew and H. S. Baird, "BaffleText: a Human Interactive Proof," *Proc., 10th SPIE/IS&T Document Recognition and Retrieval Conf. (DRR2003)*, Santa Clara, CA, January 23-24, 2003.
- [CBF01] A. L. Coates, H. S. Baird, and R. Fateman, "Pessimial Print: a Reverse Turing Test," *Proc., IAPR 6th Intl. Conf. on Document Analysis and Recognition*, Seattle, WA, September 10-13, 2001, pp. 1154-1158.
- [HB01] N. J. Hopper and M. Blum, "Secure Human Identification Protocols," In: C. Boyd (Ed.) *Advances in Cryptology, Proceedings of Asiacrypt 2001*, LNCS 2248, pp.52 -66, Springer-Verlag Berlin, 2001
- [LABB01] M. D. Lillibridge, M. Abadi, K. Bharat, and A. Z. Broder, "Method for Selectively Restricting Access to Computer Systems," U.S. Patent No. 6,195,698, Issued February 27, 2001.
- [LPRS85] G. E. Legge, D. G. Pelli, G. S. Rubin, & M. M. Schleske, "Psychophysics of Reading: I. Normal Vision," *Vision Research*, Vol. 25, No. 2, pp. 239-252, 1985.
- [MM03] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," *Proc., IEEE CS Society Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, June 16-22, 2003.
- [NS96] G. Nagy and S. Seth, "Modern optical character recognition." in *The Froehlich / Kent Encyclopaedia of Telecommunications*, Vol. 11, pp. 473-531, Marcel Dekker, NY, 1996.

- [Pav00] T. Pavlidis, "Thirty Years at the Pattern Recognition Front," King-Sun Fu Prize Lecture, 11th ICPR, Barcelona, September, 2000.
- [RNN99] S. V. Rice, G. Nagy, and T. A. Nartker, *OCR: An Illustrated Guide to the Frontier*, Kluwer Academic Publishers, 1999.
- [RJN96] S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The Fifth Annual Test of OCR Accuracy," ISRI TR-96-01, Univ. of Nevada, Las Vegas, 1996.
- [SCA00] A. P. Saygin, I. Cicekli, and V. Akman, "Turing Test: 50 Years Later," *Minds and Machines*, 10(4), Kluwer, 2000.
- [SN05] P. Sarkar & G. Nagy, "Style Consistent Classification of Isogenous Patterns," *IEEE Trans. on PAMI*, Vol. 27, No. 1, January 2005.
- [SSB03] P. Y. Simard, R. Szeliski, J. Benaloh, J. Couvreur, I. Calinov, "Using Character Recognition and Segmentation to Tell Computer from Humans," Proc., IAPR Int'l Conf. on Document Analysis and Recognition, Edinburgh, Scotland, August 4-6, 2003.
- [Tur50] A. Turing, "Computing Machinery and Intelligence," *Mind*, Vol. 59(236), pp. 433-460, 1950.
- [VN05] S. Veeramachaneni & G. Nagy, "Style context with second order statistics," *IEEE Trans. on PAMI*, Vol. 27, No. 1, January 2005.