A Highly Legible CAPTCHA that Resists Segmentation Attacks

Henry S. Baird and Michael A. Moll and Sui-Yu Wang

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

E-mail: `baird@cse.lehigh.edu`, {`mam7`|`syw2`}`@lehigh.edu`
URL: `www.cse.lehigh.edu/~baird`

Abstract. A CAPTCHA which humans find to be highly legible and which is designed to resist automatic character–segmentation attacks is described. As first detailed in [BR05], these 'ScatterType' challenges are images of machine-print text whose characters have been pseudorandomly cut into pieces which have then been forced to drift apart. This scattering is designed to repel automatic segment-then-recognize computer vision attacks. We report results from an analysis of data from a human legibility trial with 57 volunteers that yielded 4275 CAPTCHA challenges and responses. We have located an operating regime—ranges of the parameters that control cutting and scattering—within which human legibility is high (better than 95% correct) even though the degradations due to scattering remain severe.

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 PessimalPrint [CBF01] and BaffleText [CB03], Paypal's CAPTCHA [Pay02], Microsoft's CAPTCHA [SSB03], and Lehigh's ScatterType [BR05]. As discussed more fully in [BR05], fully or partially successful attacks on some of these CAPTCHAs have been reported. We and other CAPTCHA researchers 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. These observations motivated us to investigate CAPTCHAs which resist character–segmentation attacks. In [BR05] we first described the ScatterType CAPTCHA, in which each character image is fragmented using horizontal and vertical cuts, then the fragments are forced apart until it is no longer straightforward automatically to reassemble them into characters. 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 to ScatterType today or for the forseeable future. However, this is a conjecture that must be tested (see the section on Future Work).

We do not apply image degradations such as blurring, thinning, and additive noise (cf. [Bai02]) so that will not obscure style-specific shape minutiae in the fragments such as stroke width, serif form, curve shape, which we speculate may account for the remarkable human legibility of these pervasively damaged images. Experimental data reported in [BR05] also showed that subjective ratings of difficulty were strongly (and usefully) correlated with illegibility. Since then we have carried out a systematic exploration of the legibility of ScatterType as a function of the generating parameters. The principal new result is the identification of an operating regime within which human legibility exceeds 95 per cent.

2 Synthesizing ScatterType Challenges

In this section we briefly review the generating parameters (a fuller discussion is in [BR05]). ScatterType challenges are 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. In these 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.

Cutting and scattering are applied, separately to each character (more precisely, to each character's image within its own 'bounding box'). A scaling dimension (the "base length") is set equal to the height of the shortest character in the alphabet. 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 resulting x & y cut fractions are held constant across all characters in the string, but the offset locations of the cuts are chosen randomly uniformly independently for each character.

Expansion Fraction Fragments are moved apart by this fraction of base length held constant across all characters in the string.

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

The resulting images are combined, 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.

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

Fig. 1. ScatterType parameter ranges selected for the human legibility trial.

3 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 Figure **??**.

After reading the text and typing the text in, subjects rated the "difficulty level" from "Easy" to "Impossible".

4 Experimental Results

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

Human legibility—percentage of challenges correctly read—is summarized in Figure **??**. 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 (details in [BR05]).

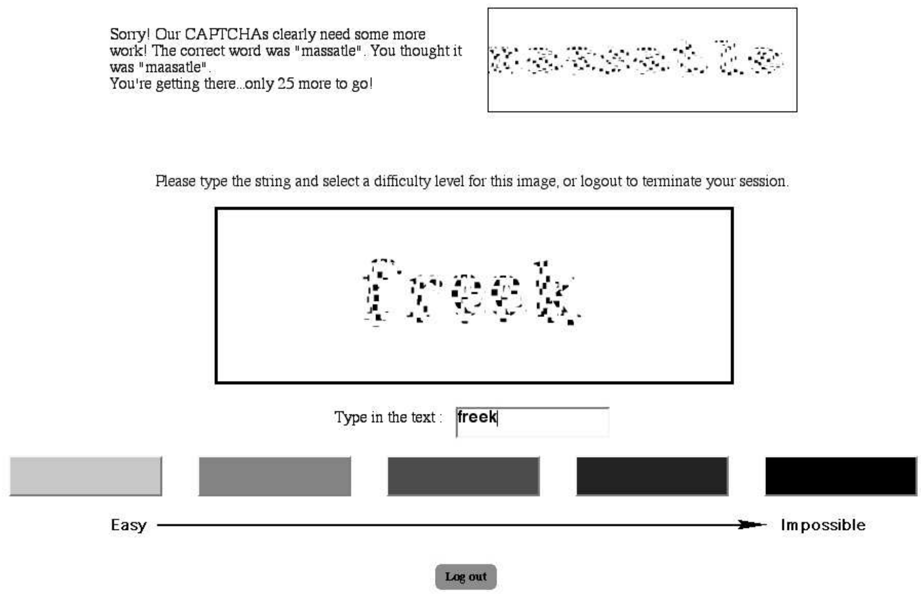

Fig. 2. An example of a ScatterType legibility trial challenge page. The Difficulty Level radio buttons (marked ’Easy’ to ’Impossible) were colored Blue, Green, Yellow, Orange, and Red. The text at the top of the page refers to the previously answered challenge.

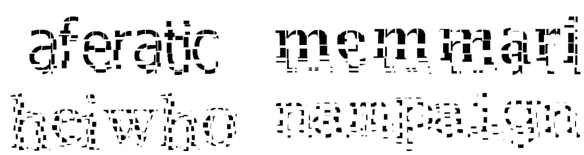

Fig. 3. ScatterType challenges rated by subjects as “Easy” (difficulty level 1 out of 5). All of these examples were read correctly: “aferatic,” “memari,” “heiwho,” “hampaign.”

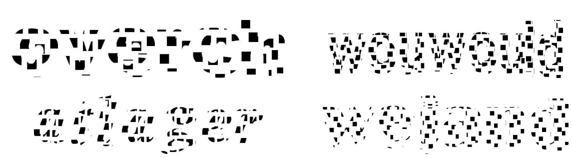

Fig. 4. ScatterType challenges rated by subjects as being of medium difficulty (difficulty level 3 out of 5). Only one of these examples was read correctly (correct/attempt): “ovorch”/”overch”, “wouwould”, “adager”/“atlager”, “weland”/”wejund”.

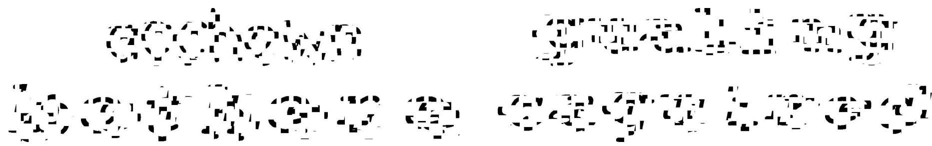

Fig. 5. ScatterType challenges rated by subjects as "Impossible" (difficulty level 5 out of 5). None of these examples were read correctly (correct/attempt): "echaeva"/"acchown", "gealthas"/"gualing", "beadave"/"bothere", "engaberse"/"caquired"

	Difficulty Level					
	ALL	1	2	3	4	5
Total challenges	4275	610	1056	1105	962	542
% correct answers	52.6	81.3	73.5	56.0	32.8	7.7

Fig. 6. Human reading performance as a function of the difficulty level that the subject selected.

5 A Highly Legible Regime

We have systematically explored the improvements in legibility that can be expected from judicious choices of generating parameters (distributions that control cutting and scattering). We began our project with 4275 ScatterType challenges collected in the human legibility trial. The overall legibility of that set (the fractions of challenges read and typed correctly) was 53%.

We used Tin Kam Ho's Mirage (http://cm.bell-labs.com/who/tkh/mirage/index.html) data analysis tool. For each challenge, we loaded the generation input parameters, the typeface used, the true word, the word guessed by user, the time taken by user to enter the guess, and the user's rating of subjective diffi culty. We examined histograms and scatter plots (colorcoded by subjective diffi culty if read correctly, with black indicating a mistake) of many single and paired features, looking for strong correlations with either objective or subjective diffi culty.

One of the fi rst features examined was the cutting fraction (set equal in both x and y directions), which had been coarsely discretized as either 0.25, 0.32, or 0.40. The cutting fraction determines the size of the rectangular blocks each of the characters bounding boxes are cut into. Therefore a smaller cutting fraction will result in more cuts and more boxes which would seem to imply the smaller the cut fraction, the more diffi cult the challenge should be to read. We created a Mirage histogram (Figure **??**) with the vertical cut fraction on the X axis: our hypothesis was confi rmed since for the three distributions of vertical cut fraction 0.25 was the only one to have more illegible than legible samples.

We then created a scatter plot (Figure **??**) with the mean horizontal scatter distance on the x-axis and the mean vertical scatter distance on the y-axis. These features determine how far each row of fragments (as created by the cutting fraction described above)

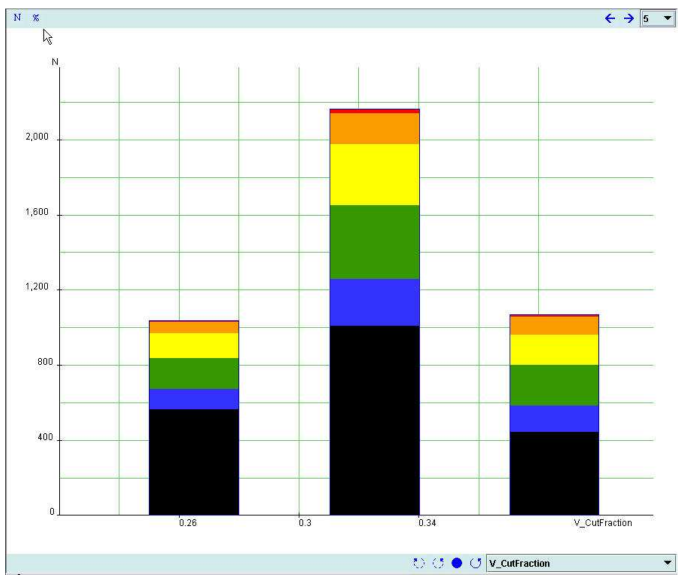

Fig. 7. Mirage histogram of difficulty levels (black marks mistakes) as a function of the Cut-Fraction parameter. The value 0.25 was the only one to have more illegible than legible samples. Black indicates a reading mistake: for legible samples, the colors red, orange, yellow, green, and blue indicate five subjective difficulty levels from "impossible" to "easy".

is displaced. The overall displacement is a positive random number that is distributed normally with a mean and standard error. In this experiment we are considering just the means which range between 0.0 and 0.40 horizontally and 0.0 and 0.20 vertically. The scatter plot in Mirage strongly indicates a higher concentration of legible challenges in the lower left hand part of the graph near the origin. Without normalizing the scales, we initially estimate best performance would result by classifying all instances within an Euclidean distance of 0.25 from the origin as legible.

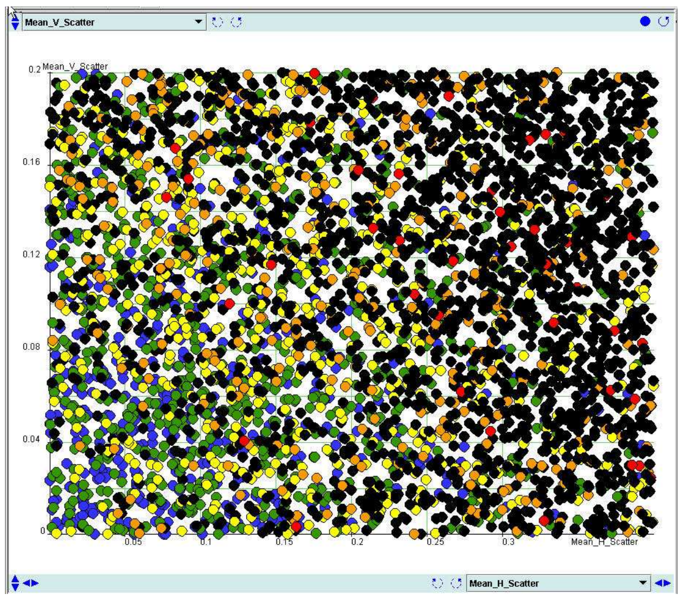

Fig. 8. Mirage scatter plot of the Mean Horizontal Scatter (X-axis) versus Mean Vertical Scatter (Y-axis) parameters. Legible samples clustered strongly near the (0,0) origin. Black indicates a reading mistake: for legible samples, the colors red, orange, yellow, green, and blue indicate five subjective difficulty levels from "impossible" to "easy".

Further exploration did not reveal any other features or pairs of features with strong correlation (positive or negative) to legibility. Two other features that we examined closely (though not within Mirage) are the font and character sets. As shown in an earlier analysis [BR05], four fonts perform significantly worse than the rest, and some characters were confused more frequently than others. The first step we took toward locating a high-legibility regime was to limit the mean scatter distances (since those parameters appeared to show the strongest correlation to legibility in our analysis using Mirage). Consider parameter d, the Euclidean distance of an instance from the origin of the scatter plot (Figure **??**) of mean horizontal scatter distance versus mean vertical scatter distance. Our initial estimate of setting $d < 0.25$ resulted in a 25% increase (Figure **??**), while still correctly classifying over one quarter of the challenges.

We then removed all cut fraction values equal to 0.25 for the reasons described above. These results, however, sharply reduced the set of challenges classified while improving legibility only slightly (Figure **??**). However the evidence of worsening performance when it was equal to 0.25 convinced us to omit this value of the parameter. Our next step was to begin removing fonts and characters that did not perform well in the trial. However, the analysis of font pruning in [BR05] showed that removing the four worst fonts resulted in positive but insignificant increases in performance at all subjective difficulty levels, especially for the two easiest levels. We repeated the analysis of removing fonts, in combination with the reduction of the cut fraction and scatter distances and verified that it did not have any correlation to improving legibility (Figure **??**). Thus we guessed that pruning fonts was unlikely to help. (Later, after pruning the worst performing characters, this hunch proved correct: pruning fonts in addition caused a *loss* of legibility of four per cent.)

In the preliminary analysis in [BR05], removing the five characters with the highest ”confusability”(’q’, ’c’, ’i’, ’o’, and ’u’) brought us rapidly to above 90%. Combined with our new restrictions, we achieved a legibility close to 93% (Figure **??**).

From this analysis we concluded that restricting mean scatter distances and pruning the worst performing characters both are strongly positively correlated with legibility, while using larger cut fraction can be somewhat useful when combined with other policies. Removing poorly performing fonts however seem to offer little benefit in increasing legibility (at least in our ”simpler” parameter space).

We continued to experiment with features to see if it would be possible to drive the legibility any higher. First we removed the next 3 worst performing characters (’z’, ’j’ and ’h’) and set $d < 0.15$ and removed cut fractions = 0.25 and increased legibility to 94.26% for 115 instances. Removing the next three most confused characters (’f’, ’n’ and ’l’) improved legibility to 95.00%, but for only 38 instances.

Taking another approach, we return to the original 5 characters removed and instead continue to decrease the d threshold to 0.1 and manage to increase legibility even further, and for more correctly classified instances than above, reaching legibility of 97.5

Obviously, a more systematic and careful study of the confusability of characters is necessary to determine which have the greatest detrimental effect on legibility, but we have shown that through removing a small subset of easily confusable characters and manipulating the values two parameters from the original trials, legibility could be raised with confidence to above 95%.

6 A Negative Result on Image Complexity

We also investigated one way to construct classifiers for legibility in spaces determined by features that can be extracted from the images of the challenges *after* they are generated. We tested the ’Perimetric Image Complexity’ metric that has been reported to be correlated negatively with legibility in the BaffleText trial [CB02]. But, as we will briefly report, this image metric failed to predict illegibility of ScatterType challenges.

Perimetric Image Complexity is an easily computed feature of any bilevel (black and white) image, as the ratio of the square of the perimeter over the black area, where

d	Cut Fraction	Fonts Removed	Chars Removed	Legibility	Correct Instances
< 0.25	0.25 - 0.40	None	None	0.697	1656
< 0.20	0.25 - 0.40	None	None	0.755	1309
< 0.15	0.25 - 0.40	None	None	0.815	809
< 0.25	0.32 - 0.40	None	None	0.715	1278
< 0.20	0.32 - 0.40	None	None	0.761	1001
< 0.15	0.32 - 0.40	None	None	0.814	613
< 0.25	0.32 - 0.40	4 Worst	None	0.744	1074
< 0.20	0.32 - 0.40	4 Worst	None	0.780	893
< 0.15	0.32 - 0.40	4 Worst	None	0.813	503
< 0.25	0.32 - 0.40	None	Q, C, I, O, U	0.788	305
< 0.20	0.32 - 0.40	None	Q, C, I, O, U	0.840	226
< 0.15	0.32 - 0.40	None	Q, C, I, O, U	0.929	143
< 0.10	0.32 - 0.40	None	Q, C, I, O, U	0.975	78

Fig. 9. Parameter ranges used to locate a high-legibility regime. d = Euclidean distance of an instance from origin of plot of mean horizontal scatter distance versus mean vertical scatter distance.

the perimeter is the length of the black/white boundary in pixels. High values correlate positively with fragmentation. In ScatterType we observed many cases where a word image was cut into a great number of pieces and yet remained legible. These cases were numerous enough to vitiate the utility of this metric to predict legibility.

7 Generating New Trials

A first step toward conducting another experiment on the human legibility of these images is to generate new trials with a parameter space constrained by our findings from the first experiment. Words containing the five most confused characters from the first trial were removed and the range for the cut fraction was reduced to 0.32 to 0.40. This was done because the smaller the cut fraction, the more blocks each character is cut into, and in the first experiments this corresponded to increasing difficulty. Also, all parameters that had been coarsely discretized in the first experiment were now more finely distributed (the number of levels for each parameters was increased to 100).

We first attempted to create trials of four different complexity levels, differentiated solely by the scatter distances. This created four classes of trials, labeled as too hard, hard, medium and easy. Upon inspecting the images generated from these parameters, a clear, incremental increase in difficulty was obvious across all four classes, however all of the classes seemed uniformly more difficult than anticipated. The easy class was expected to be almost trivial to read, yet from simply looking at those trials, it was obvious we would have to be very optimistic to expect the legibility of those trials to be over 90

Realizing that simply limiting the scatter distances from the original experiment was simply not enough to raise legibility as high as we hoped, we experimented with creating two more simpler classes, labeled as simple and trivial, by altering the parameters for expansion fraction and cut fraction. In general, the larger the cut fraction becomes, the fewer cuts that are placed in the character, and this should typically result in more legible images, as long as the expansion fraction is also not too large. As expected, the resulting class labeled simple was much easier to read, primarily because of fewer cuts being made to the character, and the class labeled trivial, was very near to the original plain text.

8 Discussion of Sample Images

The following six images illustrate the six subranges of parameters that we chose after analysis of the first experiment. We have named these classes "trivial" (Figure **??**), "simple" (Figure **??**), easy (Figure **??**), "medium hard" (Figure **??**), "hard" (Figure **??**) and "too hard" (Figure **??**). These names are to some extent arbitrary, but they capture our intuition about legibility within each subrange. This is a step towards understanding the ScatterType parameter space well enough to allow us to generate challenges in real time possessing a specified difficulty. In the following six examples, the true word is "telghby" and the font is Courier New Bold.

Fig. 10. A "Trivial" example, generated using a large cut fraction, a small expansion fraction, and no overlap due to the character separation parameter. It is indeed highly legible, so much so that some human readers might not suspect that it was a test of skill.

Fig. 11. A "Simple" example, generated with a cut fraction value that allows roughly two or three cuts and a slightly larger expansion fraction than "trivial" cases. The characters are also slightly less separated.

As these Figures illustrate, from case to case there is a gradual but perceptible increase in difficulty of these images. One potential problem with all six of these particular examples is that it does not seem difficult to segment the characters using vertical cuts in large white spaces: of course this could make them more vulnerable to attack, regardless of the degradation of the individual characters.

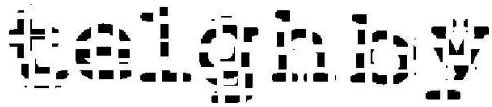

Fig. 12. An "Easy" example, generated with an expansion fraction greater than for "simple" cases: but it is still easy to segment characters using vertical strokes within wide white space channels. Note that the base of the letter 'h' is starting to merge so that it begins to resemble the letter 'b': but we believe that for most readers it will be obvious that they remain distinct characters.

Fig. 13. A "Medium Hard" example, generated using nearly the same parameters as in the "easy" cases. The principal change is an increase in the scatter distance, which in this example degrades legibility noticeably compared to Figure **??**.

Fig. 14. A "Hard" example, generated using the same parameters as "medium hard" cases, except that scatter distance has been increased. The letter 't' that starts the word is now nearly obliterated. We can still distinguish 'h' from 'b' but it is now difficult to tell which is which.

Fig. 15. A "Too Hard" example, generated using an even larger scatter distance than for the "hard" cases. At this level of difficulty, words often become illegible. Note that the letter 'b' no longer seems to have an appropriate height.

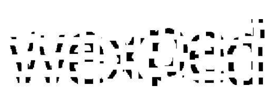

Fig. 16. The correct word is "wexped". This image has been generated using "easy" parameters but it's not highly legible. The cause appears to be small character separation, especially between 'e','x' and 'p'. Without knowing the word, it seems difficult to recover the 'x'. This illustrates the difficulty of achieving 100% legibility within the current ScatterType parameter space.

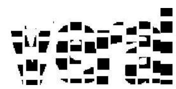

Fig. 17. The correct word is “veral”. As in Figure **??** above, it has been scattered using “easy” parameters, but in a different font. Despite small character separation it isn’t as difficult to segment as the prior example. This illustrates the problematic fact that font choice can dominate the effects of the scattering parameters, and in a manner that is hard to predict.

Fig. 18. The correct word is “tpassed”. This also was generated using “easy” parameters. It’s interesting to see two ’s’ characters treated so differently within the same word.

Fig. 19. The correct word is “spental”. This was also generated using “easy” parameters, but this case happens to achieve the desirable characteristic of being difficult to segment into characters. However, it is potentially ambiguous in its last three letters.

Fig. 20. The correct word is “neved”. It is generated using “easy” parameters, but characters are easier to segment than the case in Figure **??**. Note that each ’e’ is rendered quite differently, and ’n’ seems implausibly “mirrored.”

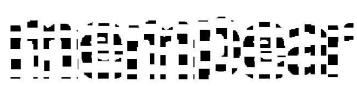

Fig. 21. The correct word is “mempear”. Generated using easy parameters, it is difficult to segment, but not because of small or negative character separation. Here, it’s due to large expansion fraction and scatter distance operating within each character.

Fig. 22. The correct word is “wested”. Generated using “medium hard” parameters, the larger scatter distance nearly destroys the legibility of the ’s’. Even small increases in parameters can have large effects.

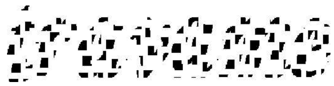

Fig. 23. The correct word is "travame". It was generated using only "medium hard" parameters however, due to interactions with the chosen font, it is uncommonly difficult to read (and to segment). This is another illustration of the interactions between scatter parameters and font which are difficult to predict and control.

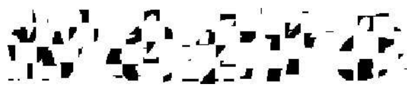

Fig. 24. The correct word is "wezre". Generated using "too hard" parameters, it is for the most part satisfactorily illegible. However it would not perhaps be difficult to segment.

We have also selected examples that illustrate instructive and problematic aspects of our approach: we discuss them below.

After generating 100 sample images for each class and viewing them, we are convinced that it will be necessary to give more careful consideration to the role that font choice plays in legibility. After the fi rst experiment, we concluded that the effect that the worst performing fonts had was greatest on those images generated with the highest subjective diffi culty and for the more legible trials, the choice of font did not play as large a part in determining subjective diffi culty. While this still appears to hold, it is not obvious that the least confused fonts actually do enhance legibility across all classes of parameters used, as seen in (Figure **??**), where using a subjectively easy font makes a word generated with the "too hard" parameters almost legible.

We have seen a great deal of evidence that ScatterType is capable of generating cases where automatic segmentation into characters would be highly problematic, while the images remain legible. This desirable property is the result of two factors: small or negative character separation of course, but also importantly large scatter distances and expansion fractions. By judicious choice of parameters we now believe we can generate a high fraction of cases with this property, but we do not yet fully understand how to guarantee it in all cases.

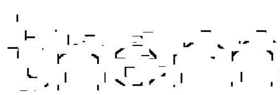

Fig. 25. The correct word is "thern". Generated using "too hard" parameters, it is indeed difficult to read, but easier than most other words generated with same parameters. Even in the more difficult regions of the parameter space, the font chosen can make a large difference in legibility.

9 Discussion and Future Work

A systematic analysis of the first ScatterType human legibility trial data has identified an operating regime—a combination of restrictions placed on generating parameters and pruning of the character set—which achieves legibility better than 95%. Within that regime we can pseudorandomly generate many millions of distinct ScatterType challenges. But the correlation of the generating parameters with these desirable properties is weak and we have nearly exhausted our experimental data in locating this regime. Future work to refine the characterization of this regime must await future legibility trials, if only to replenish the data set.

We also hope to investigate a related question: 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 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.

One reviewer suggested that the Gestalt laws of continuity (of, *e.g.*, straight and curved lines perceived as continuos inspite of breaks) may go far to explain the point of collapse of legibility. This deserves careful analysis.

Another reviewer suggested that since certain characters (*e.g.* 'c', 'e', and 'o') are more vulnerable to ScatterType degradations, they should be generated with restricted range of parameters. This technique might alleviate the problem of generating a sufficient number of nonsense words within a pruned alphabet.

Of course every CAPTCHA including ScatterType must be tested systematically using the best available OCR engines, and should be offered to the research community for attack by experimental machine vision methods.

10 Acknowledgments

We are grateful especially to Terry Riopka, who implemented the first version of ScatterType and ran its first human legibility trial.

We are also 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 warmly thank the nearly 60 people—students, faculty and staff in the Computer Science & Engineering Dept of Lehigh University, and technical staff at Avaya Labs Research—for volunteering their time for the legibility trial.

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," Dept. of Computer Science, Carnegie-Mellon Univ., `www.captcha.net`, and personal communications, November, 2000.

[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.

[BR05] H. S. Baird and T. Riopka, “ScatterType: a Reading CAPTCHA Resistant to Segmentation Attack,” *Proc., IS&T/SPIE Document Recognition & Retrieval XII Conf,*, San Jose, CA, January 16–20, 2005.

[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, “Pessimal 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 Identifi cation Protocols,” In: C. Boyd (Ed.) Advances in Crypotology, 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.

[Pay02] PayPal Captha, on display starting in 2002 at `www.paypal.com`.

[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 Classifi cation 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.