BaffleText: a Human Interactive Proof

Monica Chew[a] and Henry S. Baird[b]

[a]Computer Science Division, U.C. Berkeley, Berkeley, CA, USA
[b]Statistical Pattern & Image Analysis, Palo Alto Research Center, Palo Alto, CA, USA

ABSTRACT

Internet services designed for human use are being abused by programs. We present a defense against such attacks in the form of a CAPTCHA (Completely Automatic Public Turing test to tell Computers and Humans Apart) that exploits the difference in ability between humans and machines in reading images of text. CAPTCHAs are a special case of 'human interactive proofs,' which work in a broad class of security protocols that allow people to identify themselves as members of given groups. We point out vulnerabilities of reading-based CAPTCHAs to dictionary and computer-vision attacks. We also survey the literature on the psychophysics of human reading, which suggests fresh defenses available to CAPTCHAs. Motivated by these considerations, we propose BaffleText, a CAPTCHA which uses non-English pronounceable character strings to defend against dictionary attacks, and Gestalt-motivated image-masking degradations to defend against image restoration attacks. Experiments on human subjects confirm the human legibility and user acceptance of BaffleText images. We have found an image-complexity measure that correlates well with user acceptance and assists the generation of challenges to fit the ability gap. Recent computer-vision attacks, run independently by Mori and Malik, suggest that BaffleText is stronger than two existing CAPTCHAs.

Keywords: Human Interactive Proofs (HIPs), Completely Automatic Public Turing test to tell Computers and Humans Apart (CAPTCHAs), psychophysics of reading, optical character recognition (OCR), Gimpy, PessimalPrint, BaffleText, Turing tests

1. INTRODUCTION

We present BaffleText, a scheme for distinguishing between humans and computers in an online environment in order to block abusive automatic transactions. We describe experiments which suggest that BaffleText is well tolerated by human users and more resistant to computer vision attacks than previous schemes.

A Human Interactive Proof (HIP) is a proof that a human can construct with no special equipment, but which a machine cannot easily construct.[1,2] HIPs work in a broad class of challenge-response protocols which allow an unaided human to authenticate herself as a member of a given group, such as humans or adults. Such proofs must resist passive attacks: any party that sees the proof should be unable to falsely generate a proof of membership. A CAPTCHA, **C**ompletely **A**utomatic **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part, is a HIP whose purpose is to distinguish humans from computers. CAPTCHAs have the attractive property that they do not have to be as computationally intractable as traditional cryptographic problems — if an adversary can hire a human to take the test more cheaply than it would cost to break the test computationally, the test is secure enough.

BaffleText exploits the large gap in ability between humans and machines in reading images of text and is thus a reading-based CAPTCHA. All CAPTCHAs in use today that are known to us are reading-based.[2]

Most of this research was done while the first author was a summer intern at PARC. The first author received additional support from a National Defense Science and Engineering Fellowship, and later on from the NSF ITR grant 0122599. This paper does not necessarily reflect the position of the funding sponsors.
M.C.: E-mail: mmc@cs.berkeley.edu
H.S.B.: E-mail: baird@parc.com, Telephone: (650) 812-4481, Fax: (650) 812-4374

Figure 1: An example of the PessimalPrint CAPTCHA

Figure 2: An example of the original Gimpy CAPTCHA

1.1. Web-Security Motivations

The proliferation of publicly available services on the Internet is a boon for the community at large, but unfortunately it has invited new abuses. Programs ('bots', 'spiders') are being written to steal services and conduct fraudulent transactions. Some examples:

- Software pirates create many free online accounts and then use them to distribute stolen copyrighted material.[3]
- Recommendation systems are vulnerable to artificial inflation or deflation of ratings. For example, E-bay, a high-traffic auction website, allows its users to rate buyers and sellers on the basis of how well they complete transactions.[4] Unscrupulous sellers rate themselves positively, thousands of times automatically, in order to hoodwink buyers into believing that they are trustworthy.
- Spammers register free e-mail accounts offered by such services as Hotmail in large numbers and use them to send unsolicited email.[5]

These are just a few examples of actions which are tolerable when performed in small numbers , but become abusive when executed many times automatically.[6,7]

2. TEXT-BASED CAPTCHAS

2.1. Reading-based CAPTCHAS

All CAPTCHAs presently in commercial use exploit the ability of people to read images of text more reliably than optical character recognition (OCR) and other machine vision systems. Their challenges are created as

Figure 3: An example of EZ-Gimpy, the simplified Gimpy CAPTCHA currently in use on Yahoo!

follows: pick a word, pick a typeface, render the word using the typeface into an image, and degrade the image. The choices of word, typeface, and degradation must yield images which are easy for humans to recognize but baffling to all OCR systems. Then, if a subject can correctly transcribe (read and type in) the word in the image, the subject may be judged to be human, not a machine. Key research questions, which we consider here, include:

- What are the conditions under which human reading is peculiarly robust? What does the literature on the psychophysics of human reading suggest?
- What do computer vision, pattern recognition, and document image analysis suggest are the most intractable obstacles to machine reading?
- Where, quantitatively, are the margins of good performance located, for humans and for machines?
- Having chosen one or more of these ability gaps, how can we reliably generate an effectively inexhaustible supply of distinct challenges that lie **strictly within** the gaps?
- Will people tolerate such tests? Will they be able to read the text images without undue difficulty?
- How well does a CAPTCHA resist attack by present-day OCR and computer-vision methods?

2.2. The EZ-GIMPY CAPTCHA

Yahoo! uses a CAPTCHA called EZ-Gimpy, developed at The School of Computer Science at Carnegie-Mellon University, to protect a against automatic registration of free email accounts[6] . The EZ-GIMPY lexicon consists of 850 English words; these are rendered in various FreeType fonts and degraded using the Gimp, an image processing tool.[8,9] EZ-Gimpy's image degradations include background grids and gradients, non-linear deformations, blurring, and additive pixel noise.

Greg Mori and Jitendra Malik of the Computer Science Division at U.C. Berkeley describe an attack on EZ-Gimpy, using lexical knowledge and 'generalized shape contexts,' which enjoyed a success rate of 83%.[10] Generalized shape contexts are used to find candidates similar to the target shape, and then the lexicon is used to prune the tree of candidate words.

2.3. The PessimalPrint CAPTCHA

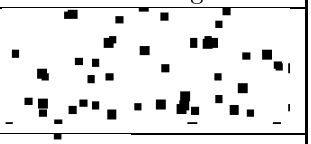
Another early example of a reading-based CAPTCHA is PARC/UCB's PessimalPrint (Figure 1).[11] Its lexicon contains only common English words, so as not to penalize young or non-native English readers. The words are between 5 and 8 characters long, with no ascenders (except for *i*) or descenders, to defend against character shape-coding OCR.[12] The motive for the length restriction is that short words are more subject to brute-force template matching attacks, and long words are more vulnerable to word-shape recognition (and more burdensome for humans). This approach yields a small, fixed lexicon. There are over a thousand words in the standard Unix dictionary that meet the length and ascender-descender restrictions, but many of them are uncommon: in the end, PessimalPrint used only 70 words.

Unfortunately, such a lexicon is far too small. If an attacker merely tried one word from it at random, the CAPTCHA would break with probability 1/70. This attack requires no computation and is easily automated, so the cost of a single successful authentication is low.

PessimalPrint uses the Baird degradation model to degrade the word images, simulating physical defects caused by copying and scanning of printed text.[13] Systematic testing using synthetic images generated by this model has located the margins of good performance for several OCR technologies.[11,14]

We invited Mori and Malik to attack PessimalPrint, using the same parameters as in the EZ-Gimpy attack. On a set of 10 PessimalPrint images using the Courier font, their attack produced 4 correct answers. For 4 of the remaining images, the correct word was in fourth place or higher in the ordered list of candidates. The Courier font used in the attack is not an exact match for the original PessimalPrint font. In another set of 17 PessimalPrint images, in which the fonts were randomly chosen and not identified, the Mori-Malik attack failed. So, when the font is approximately known, the Mori-Malik attack can break PessimalPrint 40% of the time. Scaling up to a large number of known fonts, though not yet attempted, might be fairly successful too.

Table 1: Examples of using a mask for degradation

	word image	mask image
type		
add		
subtract		
difference		

2.3.1. User Acceptance

It is critically important that human users not find a CAPTCHA annoying or excessively difficult. Yahoo! has 98 million registered users, and Hotmail has 100 million; so even if 99.999% percent of the population doesn't complain, a CAPTCHA administrator could still face on the order of 1000 complaints.[3,15] The first Yahoo! CAPTCHA, called Gimpy (Figure 2), triggered user complaints necessitating its replacement by the simpler EZ-Gimpy (Figure 3).

2.4. Where OCR Technology Fails

One possible attack on reading-based CAPTCHAs is to apply image-restoration operations before OCR. Almost all current techniques for image restoration focus on removing the effects of blurring, thresholding, and per-pixel (e.g. additive) random noise (such as are relied on in PessimalPrint). However, there are other classes of degradations, such as occlusion or interference by random shapes, which obliterate parts of the image. Image restoration methods cannot replace these missing parts without prior knowledge of the occluding shapes. However, humans are remarkably good at recognizing an entire shape or picture in spite of incomplete, sparse, or fragmented information: this is an example of a 'Gestalt perception' ability.

We have therefore chosen to attempt to generate degradations which exercise Gestalt abilities. Table 1 exhibits three types of mask operations: addition, subtraction, and difference.* Our use of these masks is described later in Section 4.

3. THE PSYCHOPHYSICS OF READING

We have surveyed the literature on the psychophysics of normal human reading in the English language, with the aim of identifying properties of images of printed text which (a) affect human legibility as measured by accuracy of transcription and which (b) affect the difficulty of the reading experience in terms of time and effort expended, or user annoyance.

*For black (1) and white (0) images, adding is equivalent to Boolean OR, subtracting to NOT-AND, and difference to XOR.

3.1. Optimal Presentation of Text

Humans can read best when the subtended angle from the eye to the character height is 0.3-2.0°.[16] Assuming a distance from the eye to the monitor screen of about 20 inches, the optimal range of character height is from 0.2 to 0.7 inches. There is no reason not to present people with this optimal character size, since OCR programs are free to rescale images as best suits them.

Legge et al. have studied performance metrics for human reading such as the critical reading rate (the maximum rate at which humans can read with acceptable errors), but researchers must make further studies to determine whether these metrics relate to whether subjects find a reading-based CAPTCHA burdensome.[16]

3.2. Helpfulness of Linguistic Context

Reading-based CAPTCHAs can use correctly spelled words, random strings of characters, or something in between. The use of correctly spelled English words certainly assists legibility for humans (as well as for machines): studies show that human subjects can read a whole English word faster than they can read a single letter, even the first character of the word![17] Subjects also read with less effort, and perhaps with higher accuracy, if the challenges are correctly spelled English words.

We use non-English but pronounceable character strings (described in the next Section). This probably increases legibility for human users, while it alleviates the small known lexicon problem. However, this choice comes with tradeoffs. Recent research suggests that the time to recognize a visually presented word may be a function of the frequencies of orthographically similar 'stimulus' words that have recently been seen.[18] Reading error rate increases on words having at least one higher frequency neighbor. As a result, random but pronounceable words may be problematic in CAPTCHAs since people may tend to identify them as real English words (e.g, misreading 'separate' for 'seperate'). The literature does not suggest how to quantify this effect.

4. THE BAFFLETEXT SPECIFICATION

BaffleText is a reading-based CAPTCHA that uses random masking to degrade images of non-English pronounceable character strings. Each BaffleText challenge is generated as follows:

1. generate a pronounceable character string and ensure it is not in the English dictionary;
2. choose a font;
3. render the character string using the font into an image (without physics-based degradations);
4. generate a mask image (described below)
5. choose a masking operation from among 'addition,' 'subtraction,' and 'difference' (see Table 1) uniformly at random; and
6. combine the character-string image and mask image using the masking operation.

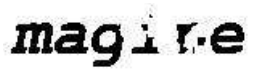
Parameters governing mask generation include:

1. Masking shape: Any combination of circles, squares, and ellipses. The minimum and maximum radius determine the size range of the shapes.
2. Minimum radius or radii (in pixels) of a masking shape: for circles, this was the ordinary radius; for squares, this was the half of the minimum allowable side length; for ellipses, this was half the major axis or half the minor axis.
3. Maximum radius or radii (in pixels) of the masking shape, similar to minimum radius.
4. Density: the fraction of black pixels in the resulting mask.

Pronounceable character strings are generated by a character-trigram Markov model.[19] The strings contain only lowercase alphabetic characters and are between 5 and 8 letters long. They also are filtered so they are not English words. Preliminary findings show that the number of eligible strings is linear in the number of bytes generated by the Markov model, so we can add strings as needed.

Examples of BaffleText challenges can be seen in Table 2. Section 6 explains the image complexity metric P^2/A.

Table 2: Examples of BaffleText

Image	word	Image	word
	obviouse, $P^2/A = 298$		quasis, $P^2/A = 280$
	alued, $P^2/A = 115$		brience, $P^2/A = 118$
	emperly, $P^2/A = 90$		finans, $P^2/A = 49$
	magine, $P^2/A = 113$		othis, $P^2/A = 14$
	ourses, $P^2/A = 113$		privally, $P^2/A = 178$
	thates, $P^2/A = 309$		publice, $P^2/A = 2900$

5. THE BAFFLETEXT EXPERIMENT

We generated BaffleText challenges and tested them on human users, as follows. First we used a tri-gram Markov model trained on the Brown corpus to generate 2758 pronounceable character strings (in production they could be generated on the fly as needed).[20] Picking fonts uniformly at random from among 72 FreeType fonts,[21] we rendered all the text strings as black and white images at a type size of 40 pixel/em. Assuming a desktop resolution of 1024x768 and a monitor height of 12 inches, these strings are 0.3 inches high and so at 20 inches viewing distance subtend about 0.9° which is within the optimal readability range of 0.3-2°. We then generated 2000 masks consisting of masking shapes chosen and placed at random. The masking shapes were combinations of squares, circles, ellipses. The parameters for generating the masking shapes (Section 4) were:

1. minimum radius or radii (in pixels) of a masking shape: 1;
2. maximum radius or radii (in pixels) of the masking shape: We tried maximum radii ranging from 5 to 15 pixels; and
3. density (fraction of black pixels): 1% to 50%.

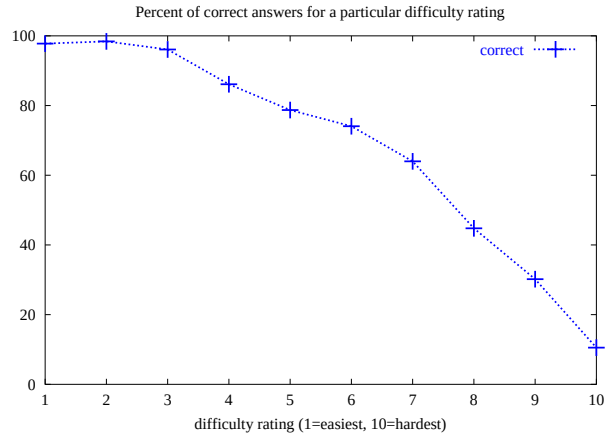


Figure 4: Percent of correct answers vs. difficulty rating

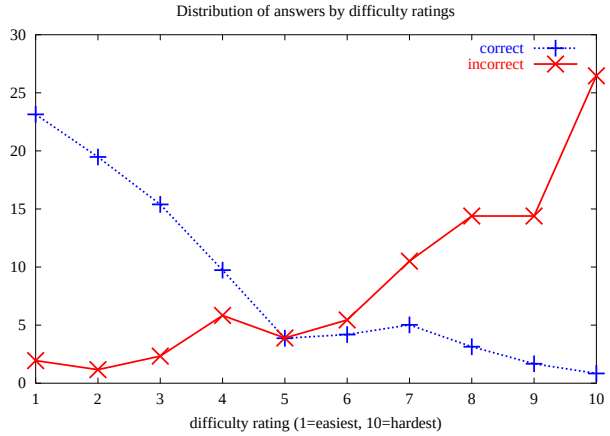


Figure 5: Distribution of answers by difficulty rating

Finally, for each image, we selected a mask operation — addition (Boolean OR), subtraction (Boolean NOT-AND), or difference (Boolean XOR) — uniformly at random. We did this twice for each clean word image. Altogether, we generated more than 5000 BaffleText challenges.

We constructed a BaffleText website and invited 33 PARC employees (including summer interns) to visit it and transcribe as many BaffleText challenges as they desired. We recorded their answers, response times, and optional comments. Response time was measured at the server side, and so included the (usually negligible) round-trip network communication time between server and client machines.

Also, subjects rated the perceived difficulty of each image, on a scale of 1-10 (10=hardest), *before* we revealed to them whether or not their answer was correct (to eliminate bias in the difficulty ratings). This was to help us understand how to generate BaffleText with low perceived as well as actual difficulty for humans (and, of course, high actual difficulty for machines).

6. RESULTS

The subjects transcribed and rated the difficulty of 1212 BaffleText challenges altogether. Their transcriptions were correct in 79% of the cases. The average response time was 6.6 seconds for correct transcriptions and 15 seconds for incorrect transcriptions. Figure 4 shows that perceived difficulty is well correlated with actual

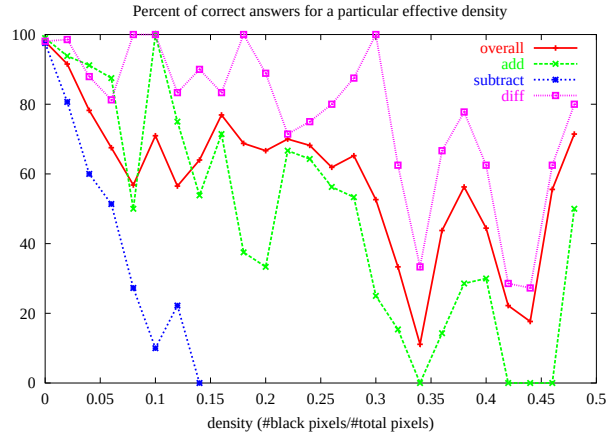


Figure 6. Accuracy vs. effective density. Since density is a continuous function, this plot buckets the density into intervals of width 0.02.

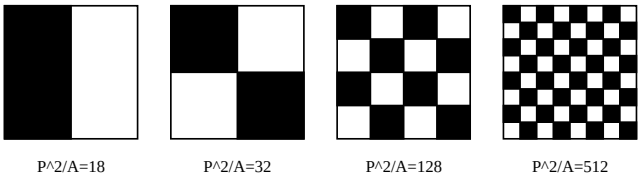


Figure 7: Same density (0.5), different complexities

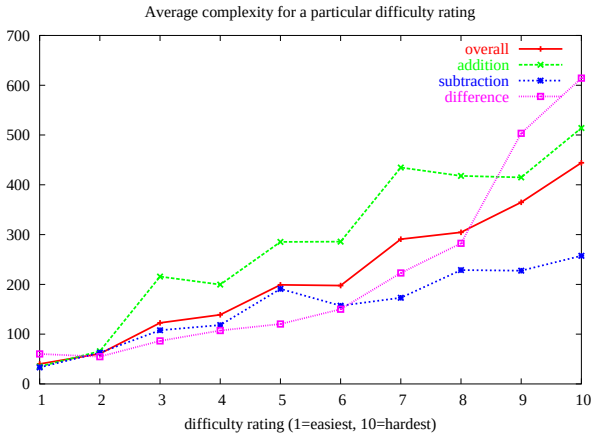


Figure 8: Complexity (P^2/A) vs. difficulty rating

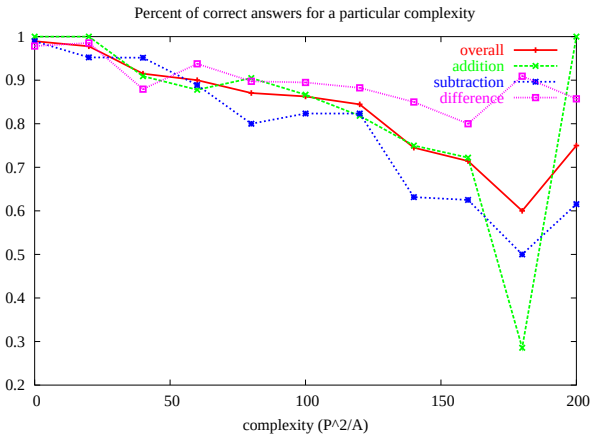


Figure 9. Accuracy vs. complexity (P^2/A). Since complexity is a continuous function, this plot buckets complexity into intervals of width 25.

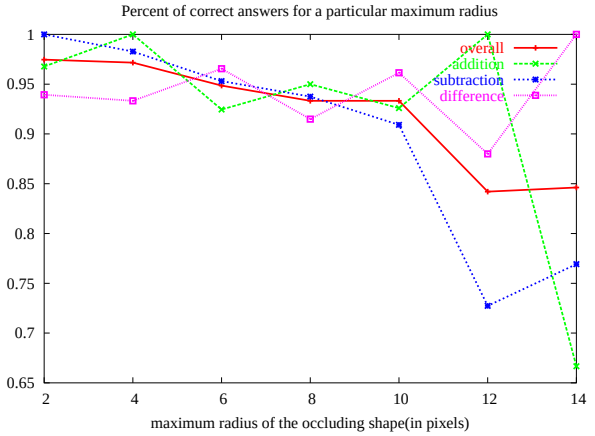


Figure 10: Accuracy vs. maximum radius of occluding shape for $P^2/A \leq 100$

difficulty. Figure 5 shows that 68% of all the correct trials were rated 4 or below in difficulty, and 67% of all the incorrect trials were rated 7 or above.

It would be highly useful to be able to predict difficulty (both actual and perceived) of a challenge at the time we generate it. In our first attempt at this, we examined the density parameter (the fraction of black pixels) of the 'effective mask': that is, the mask pixels that make some difference when applied to the original word image.[†] However, Figure 6 shows that this 'effective density' does not correlate well with objective difficulty. The reason for this becomes clear in Figure 7: the four images in Figure 7 have the same density, but each figure is more complex than the last.

Next, we tried an image metric described in the psychophysics of reading literature, *perimetric complexity*.[22] This measures the length of the boundary between black and white (the 'perimeter') squared divided by the black area, P^2/A. Perimetric complexity is unitless, independent of scaling, and additive for equi-area shapes. Table 2 shows examples of BaffleText and their perimetric complexities. As Figures 8 and 9 show, perimetric complexity is better correlated with both perceived and actual difficulty than effective density is. We can use it to allow the selection of BaffleText challenges to lie within given ranges of difficulty ratings: e.g. to achieve a difficulty rating of 4 or less, pick challenges with perimetric complexity less than 100.

The maximum radius of the occluding shape also affects the legibility of BaffleText. As radius increases, accuracy decreases for addition and subtraction, but increases for difference (Figure 10). This occurs because applying difference to large occluding shapes inverts large sections of the original image while still retaining high human readability. Subtraction and addition cause the greatest loss in human legibility with increasingly large radii, because large occluding shapes obliterate large parts of the image.

6.1. Exit Survey Results

Of the 33 participants, 18 took the exit survey on their background and reactions to the BaffleText experiment. Of the 18,

- 3 reported they would be willing to solve a BaffleText every time they sent email;
- 7 reported they would be willing, if it reduced spam tenfold;
- 14 said they enjoyed 'perfect vision' (with corrective lenses if necessary);
- 16 reported they would be willing to solve one every time they registered for an e-commerce site;
- 17 reported they would be willing, if it meant those sites had more trustworthy recommendations data; and
- all 18 reported they would be willing to solve one every time they registered for an e-mail account.

6.2. Engineering Recommendations

Our experimental results and literature survey encourage us to make specific recommendations for BaffleText engineering:

- generate images with perimetric complexities between 50 and 100, so as not to make the test too frustrating for humans; and
- use the difference masking operation, since humans can read BaffleText under difference very well, and at higher perimetric complexities than under addition and subtraction.

For BaffleText challenges conforming to these policies, our experimental data shows the human subjects answered correctly 89% of the time, averaging 8.7 seconds per trial. Also, for reading-based CAPTCHAs in general, we suggest rendering the word to fall within the optimal range for legibility, i.e., between 0.2-0.7 inches high for a sitting distance of 20 inches.

[†]For example, with addition the parts of the mask where the original clean word image is black do not matter, so we subtract the original image from the challenge image to give the effective mask image. Similarly, for subtraction we subtract the challenge from the original.

6.3. Attacking BaffleText

We have subjected BaffleText to the Mori-Malik attack in order to compare it with EZ-Gimpy and PessimalPrint.[10] In those attacks, Mori and Malik had full knowledge of both the lexicon and the font: so, to allow for straightforward comparisons, we allowed the same for BaffleText (atypically, since in normal practice both would be unpredictable). We picked 15 words from EZ-Gimpy's lexicon and applied the addition, subtraction, and difference operations to each word, picking the operation and mask uniformly at random. The effective masks had perimetric complexity between 28 to 405. The words were all rendered in the same Courier font that the Mori-Malik attack assumes. In a test of 45 images, 11% were correct (2 difference, 2 addition, and 1 subtraction). Of the images falling in our recommended range (50-100 perimetric complexity), 25% were correct. The results of the Mori-Malik attack, given full knowledge of font and lexicon, are summarized in Table 3. BaffleText resists this attack better than EZ-Gimpy and PessimalPrint, even when the attacker has full knowledge of font and lexicon.

Table 3: Results of the Mori-Malik attack on reading-based CAPTCHAs with full knowledge of font and lexicon

CAPTCHA	Success rate
BaffleText	11%
BaffleText (recommended complexity range)	25%
PessimalPrint	40%
EZ-Gimpy	83%

7. DISCUSSION

The techniques underlying BaffleText, inspired by an analysis of weaknesses of earlier CAPTCHAs, knowledge of machine vision, and guidance from the literature on the psychophysics of reading, appear promising. We have found that perimetric image complexity correlates strongly with actual difficulty (the objective error rate) people have in reading. Perceived (subjective) difficulty is also strongly correlated with perimetric image complexity. With this knowledge we can generate BaffleText challenges which are likely to be legible and well-tolerated by human users, while resisting attack by modern computer vision technology better than two other CAPTCHAs. Questions for further investigation are:

- How well (or poorly) do a wide range of existing commercial OCR machines perform on BaffleText?
- Can we develop image restoration techniques or other machine vision techniques that break BaffleText?
- Do our engineering recommendations produce BaffleText that the general public is willing to transcribe without complaint?

ACKNOWLEDGMENTS

We are grateful to Gordon Legge for advice on psychophysics of reading. Tom Breuel implemented the program used to render the word images. Kris Popat wrote the N-gram model used to generate pronounceable words. Doug Tygar and Rob Johnson contributed many useful editorial comments. Greg Mori kindly ran his and Jitendra Malik's attacks on BaffleText and PessimalPrint for us.

REFERENCES

1. N. J. Hopper and M. Blum, "Secure human identification protocols," in *Proc., ASIACRYPT '01, LNCS 224*, pp. 52–66, 2001.
2. H. S. Baird and K. Popat, "Human interactive proofs and document image analysis," in *Proc., IAPR 2002 Workshop on Document Analysis Systems*, (Princeton, NJ), August 2002.

3. U. Manber, "Large scale internet security." Invited talk, IEEE Symposium on Security and Privacy, May 2002.
4. eBay, "ebay." http://www.ebay.com.
5. Microsoft, "Hotmail." http://www.hotmail.com.
6. M. Blum, L. A. von Ahn, J. Langford, and N. Hopper, "The CAPTCHA Project: Completely Automatic Public Turing test to tell Computers and Humans Apart." http://www.captcha.net, November 2000.
7. D. P. Baron, "eBay and database protection." Case No. P-33, Case Writing Office, Stanford Graduate School of Business, 518 Memorial Way, Stanford Univ., Stanford, CA 94305-5015, 2001.
8. C. K. Ogden, "Ogden's basic English." http://www.basiceng.com, 1930.
9. P. Mattis and S. Kimball, "The GNU image manipulation program." http://gimp.org, 2002.
10. G. Mori and J. Malik, "Breaking a visual CAPTCHA," 2002. In submission to Computer Vision and Pattern Recognition 2003.
11. A. L. Coates, H. S. Baird, and R. Fateman, "Pessimal print: a reverse Turing test," in *Proc., IAPR 6th Intl. Conf. on Document Analysis and Recognition*, pp. 1154–1158, (Seattle, WA), September 2001.
12. A. L. Spitz, "Moby dick meets geocr: Lexical considerations in word recognition," in *4th International Conference on Document Analysis and Recognition*, pp. 221–232, (Ulm, Germany), August 1997.
13. H. S. Baird, "Document image defect models," in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, eds., pp. 546–556, Springer–Verlag: New York, 1992.
14. T. K. Ho and H. S. Baird, "Large-scale simulation studies in image pattern recognition," *IEEE Trans. on PAMI* **19**, pp. 1067–1079, October 1997.
15. M. Burrows, "Fighting spam with memory bound puzzles." 15 minute talk, 2002.
16. G. E. Legge, D. G. Pelli, G. S. Rubin, and M. M. Schleske, "Psychophysics of reading: I. normal vision," *Vision Research* **25**(2), pp. 239–252, 1985.
17. O. J. L. Tzeng and H. Singer, *Perception of Print: Reading Research in Experimental Psychology*, Lawrence Erlbaum Associates, Inc., 1981.
18. J. Grainger and J. Segui, "Neighborhood frequency effects in visual word recognition: A comparison of lexical decision and masked identification latencies," *Perception & Psychophysics* **47**, pp. 191–198, 1990.
19. K. Popat, D. Bloomberg, and D. Greene, "Adding linguistic constraints to document image decoding," in *Proc., 4th IAPR Workshop on Document Analysis Systems*, (Rio de Janeiro, Brazil), December 2000.
20. W. Francis and H. Kucera, "The Brown corpus of standard American English," 1961.
21. D. Turner, "The freetype project." http://www.freetype.org, 2002.
22. D. G. Pelli, C. W. Burns, B. Farell, and D. C. Moore, "Identifying letters," *Vision Research, accepted with minor revisions* , 2002.