1W-07

一筆書き CAPTCHA の問題画像と解答データの自動生成手法の検討

小松原 健† 髙谷 眞弓† 山村 明弘†

1. はじめに

スマートフォンの特徴である,画面が小さい,タッチ操作ができる点を活かした CAPTCHA として,Tsuruta らはスマートフォンに適した CAPTCHA を提案した[1]. しかし, [1]では,問題と解答データをプログラムで生成する手法について述べられていない. そのため出題できる問題の数を増やすには手間と時間が必要であった. 出題される問題の数が限られると,攻撃者がプログラムを用いて CAPTCHA を突破するためにあらかじめ解答データを準備することが簡単である. そこで,問題と解答データをプログラムにより自動生成する手法を web アプリケーションの形式で実装し,その手法について述べる.

2. 一筆書き CAPTCHA の概要

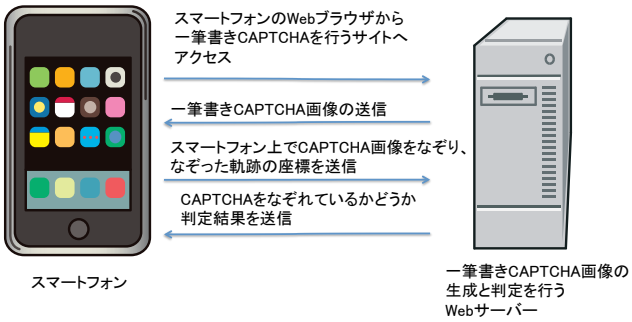

図 1 一筆書き CAPTCHA の認証手順

[1]で提案された手法を一筆書き CAPTCHA と呼ぶことにし、その認証手順を図 1 に示す.

英文字「e」の場合の例を元に認証を受理する場合, されない場合を説明する.

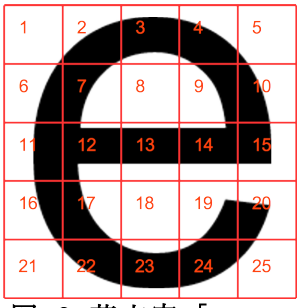

図 2 英文字「e」

まず文字画像の表示領域を格子状に区切り, それぞれに唯一の番号を与える(図 2). そして座標データを取得し, どの格子の番号に含まれるかを調べる. 例では, 12, 13, 14, …, 20 のように書き順通りの入力ならば受理される. 不受理となる場合は, 17, 23, 25, 20, …のように文字からずれた領域をなぞるデータの場合不受理となる.

3. 問題と解答データの自動生成手法

実装に使用した言語はクライアント側に JavaScript, サーバーは Java, Apache tomcat を用いている.

3.1. 問題画像の生成手法

一筆書き CAPTCHA に対する攻撃手法として, 文字認識と, ラインとレース攻撃が考えられる. そこで, 文字を歪ませる, 文字に切れ目を入れる, 画像にランダムにノイズ点を入れる対策を行った. また文字を歪ませることによりなぞる座標点が変化するため, あらかじめ正解となるデータを用意することが難しくなることが予想される. 文字は PerspectiveFilter クラス[1]を用いて, 画像の 4 隅の座標点をランダムに動かし歪めて, さらに波状に歪めている. 文字に入れる切れ目は, 画像の重心点から放射状に幅 10px の白線を 6 本入れている. 図 3 は生成した文字画像の例である.

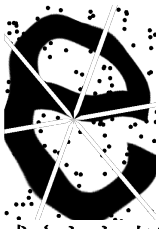

図 3 生成された文字画像

3.2. 解答データの生成手法

解答データは次の手順で生成している. 歪めてない状態の文字画像をなぞった時に通過する格子の番号と順番を, 基本解答データと呼ぶことにする.

歪めてない e を表示させ, 基本解答データを調査する(図 4 の左図の例では, [17,18,19,…,25]が基本解答データとなる). 格子の番号をn, 格子の幅をw, 一行の格子の数をtとする. 式(1)を用いて格子の番号から, その格子の中央のx,y座標点に変換する.

Study of one stroke CAPTCHA problems image and automatic method of generationg answer data using the server
Takeshi Komatsubara, Mayumi Takaya, Akihiro Yamamura
† Akita University

[1] http://www.jhlabs.com/ip/filters/

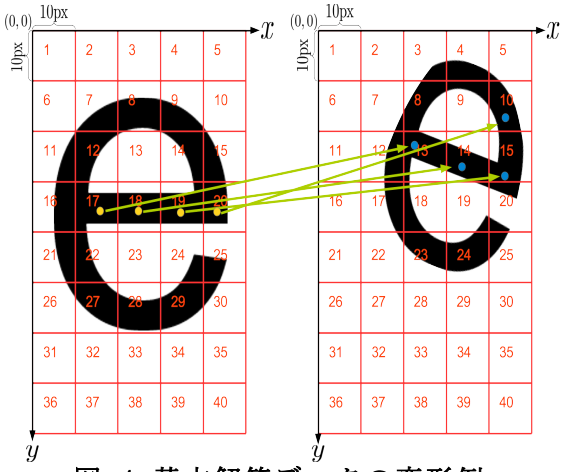

図 4 基本解答データの変形例

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} w(n \bmod t) + \frac{w}{2} \\ w\left\lfloor \frac{n}{t} \right\rfloor + \frac{w}{2} \end{pmatrix} \quad (1)$$

画像を歪める前後の画像の 4 隅の座標点を用いて式(2),(3)の射影変換式のパラメーター$a, \ldots, h$を求める．

$$X = \frac{ax + by + c}{gx + hy + 1} \quad (2)$$

$$Y = \frac{dx + ey + 1}{gx + hy + 1} \quad (3)$$

格子の中央のx, y座標点を射影変換式(2),(3)を用いて移動させる(図 4 の右図)．式(4)を用いてx, y座標点から格子の番号に変換する．

$$n = \left\lfloor \frac{x}{w} \right\rfloor + t * \left\lfloor \frac{y}{w} \right\rfloor \quad (4)$$

この結果，基本解答データ[17,18,19,…,25]から[13,14,15,10,…]が得られる(図 4 の右図)．

最後に，誤判定を減らすために，変換した書き順データの拡幅を行う．射影変換によって得られた書き順データを$L = [l_1, l_2, \ldots, l_i, \ldots, l_k]$とする．$l_i$には格子の番号が入る．この時$l_i$の 8 近傍の格子を同じ書き順の格子とする．ただし，l_1と同じ書き順の格子とするのは，l_0で指定した同じ書き順の格子と重ならない格子のみとする．$i \geq 2$のときは，l_{i-1}, l_{i-2}で指定した同じ書き順の格子と重ならない格子のみとする．例として，基本解答データを射影変換して得られた格子の番号が$L = [13,14,15,10 \ldots]$だとする．これを拡幅した例を図 5 に示す．赤数字が格子の番号，青数字が書き順を表している．青数字が複数個ある格子は複数回通過する格子である．

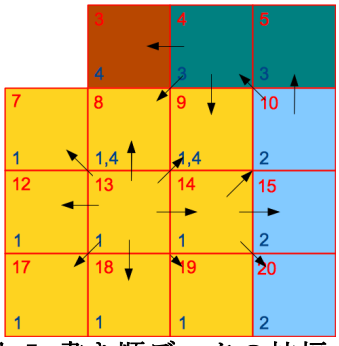

図 5 書き順データの拡幅

4. ラインとレース攻撃に対する安全性

一筆書き CAPTCHA に対する攻撃手法の一つとしてラインとレースを用いて文字をなぞる手法が考えられる．そこで，提案した手法により生成した問題画像がラインとレースを行えない画像であるかを調査した． S を 4 通りに歪めた問題画像(図 6～9)にラインとレースシミュレーターver1.1 を用いてラインとレースを行った．図 6～9 の赤矢印の位置がラインとレーサーの開始位置であり，青い長方形がラインとレーサーの停止位置である．S の書き順の終点からトレーサーをスタートさせているのは，トレーサーは一旦スタート位置から下方向に進み，黒い線を探索するからである．ここでは，S を逆の書き順で一筆書き出来た場合にトレース可能な画像である判断することにした．

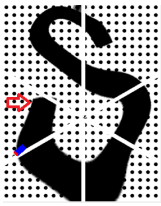

図 7 S_1

図 6 S_2

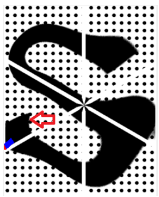

図 9 S_3

図 8 S_4

図 6～9 はトレーサーが切れ目の位置で停止しているため，トレース不可能な画像である事がわかる．この結果から画像の重心点から放射状に切れ目を入れる手法は，ラインとレース攻撃を防ぐ対策として有効であると考えられる．

5. まとめ

今回一筆書き CAPTCHA の問題と解答データをプログラムで生成する手法を web アプリケーションの形式で実装した．今後さらに，文字認識やラインとレース攻撃に対する安全性の検証を進める．

参考文献

[1] Y. Tsuruta, M. Takaya, and A. Yamamura, “CAPTCHA Suitable for Smartphones,” *Inf. Commun. Technol.*, pp. 131-140, 2013.