ランダムドットを用いた文字列CAPTCHAの提案

山場　久昭 [a)]・市元　祐弥 [b)]・油田　健太郎 [c)]・岡崎　直宣 [d)]

A Proposal of a Reading Text CAPTCHA Using Random Dot Patterns

Hisaaki YAMABA, Yuya ICHIMOTO, Kentaro ABURADA, Naonobu OKAZAKI

Abstract

According to the a growth of troubles caused by malicios programs called bots, CAPTCHA (Comletely Automated Public Turing test to tell Coputers and Humans Apart) comes to be an important roll in the current information society. CAPTCHA identifies bots from legitimate human users by requiring some questions that are easy for humans to solve but difficult for bots. However, the progress of computing technology such as OCR, bots come to be able to solve present CAPTCHAs. In order to outcome such troublesome bots, more sophisticated CAPTCHA that equips high cognitive ability like human is desired. In this paper, we propose a new CAPTCHA scheme that uses random dot patterns. Human can recognize a moving cluster filled by random dot patterns from a background filled by random dot patterns; however, loses the cluster in the background when the cluster pauses. Since image recognition by computer programs is usually carried out frame by frame, it is hard for bots to recognize a moving cluster filled by random dot patterns from a random dot pattern background. The proposed CAPTCHA requires users to answer a text that is filled by a random dot pattern and moves on a background that is also filled by another random dot pattern. Several experiments were carried out to confirm that the proposed CAPTCHA scheme has enough resistance against bots attacks using representative image recognition methods. Other experiments were also carried out to evaluate the usability of the proposed CAPTCHA scheme. The system usability scheme (SUS) was adopted in the experiments. The results of the experiments showed that the CAPTCHA scheme is usable enouth and has enoungh resistance against bots attacks.

Keywords: CAPTCHA, bot, random dot pattern

1. はじめに

近年、ボットと呼ばれる自動プログラムを用いたウェブサービスの不正利用が問題視されるようになってきている。これは例えば、無料メールサービスなどの Web サービスにボットと呼ばれる自動プログラムでアクセスさせ、用いてアカウントを大量取得した上で、それらを SPAM メールの送信に利用する、というものである。

このような問題を防止するために、CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) と呼ばれる手法を活用して、人間とボットの識別が図られている [1)]。CAPTCHA はチャレンジ/レスポンス型テストの一種であり、人間には容易に解答できるがコンピュータには判別が困難である問題を出題し、正解できた場合にその回答者を人間である、と判断する。現在、Web サービスを提供するサイトの多くが、マルウェアによる攻撃を防ぐ手段として、文字列 CAPTCHA や画像 CAPTCHA （図 1）を採用している。

[a)] 情報システム工学科助教
[b)] 情報システム工学科学部生
[c)] 情報システム工学科准教授
[d)] 情報システム工学科教授

しかし昨今、CAPTCHA をより高度化していくことが求められるようになってきている。というのも、近年の文字認識技術や機械学習の発達により、現在の主流である文字列 CAPTCHA や画像 CAPTCHA は、容易に突破されるようになってきているからである。

高度化へのアプローチの一つが、人間の高度な認知能力を必要とする CAPTCHA の導入である。高度な認知能力の模倣は、マルウェアにとって攻略が困難な課題の一つであると期待されており、近年提案される CAPTCHA には、このような高度な認知能力を利用するものが多くなっている。

そこで本論文では、人間の高度な認知能力を必要とする新たな CAPTCHA 方式として、ランダムドットパターンを用いた CAPTCHA を提案する。基本的には文字列 CAPTCHA をベースとした動的な CATPCHA であるが、ランダムドットパターンの背景の上を、やはりランダムドットパターンで描かれた、移動する文字列を判読できるか否かによって、人間かボットかを判断する。さらに、OCR 機能を備えたボットの突破率を低下させるために、画像解析を阻害するような工夫も追加している。具体的には、透過性を持ったた別のランダムドットパターンを、背景・文字列上に表示することによって、文字を抽出・判読されることへの耐性を高めている。

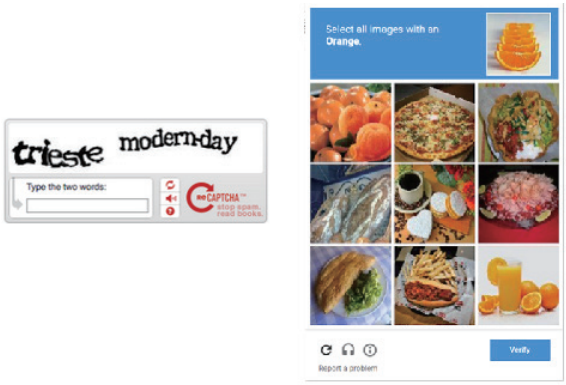

図 1. 文字列 CAPTCHA(reCAPTCHA V1) と画像 CAPTCHA(reCAPTCHA V2) の例

2. 先行研究

2.1 文字列 CAPTCHA

文字列 CAPTCHA は現在最も広く利用されている CAPTCHA である。良く知られた文字列 CAPTCHA としては Gimpy[2)]、EZ-Gimpy[2)]、r-Gimpy[3)] などがある。これらは、アルファベットと数字をランダムに並べた文字列の画像を歪ませて表示する。回答者は元の文字列を推定してテキストボックスに入力し、その入力が正しければ回答者を人間と判断する。

文字列 CAPTCHA の長所として、Web サイトに簡単に取り入れることが可能である単純さと、総当たり攻撃に高い耐性を持つことが挙げられる。短所としては、近年の OCR（光学文字認識）技術の発達により、ボットが簡単に文字を認識できてしまう可能性が高くなってしまっていること挙げられる。

2.2 ランダムドットパターン

ランダムドットパターンとは、領域内を黒か白点で埋め尽くしたものであり、黒・白いずれであるかは無作為に決められている。

渡邊はランダムドットパターンの背景上を、別のランダムドットパターンの塊が移動する時には、人間がその輪郭を知覚できるに対して、それが静止した時には識別できなくなることを示している [4)]。ゲシュタルト心理学における「共通運命の法則」によれば、人間の脳には、同時に発生し、同じように変化している複数のものをひとまりに認識する機能が備わっていることが知られている [5)]。

2.3 SNOW NOISE CAPTCHA

SNOW NOISE CAPTCHA[6)] は、ランダムドットパターンを利用した CAPTCHA である。ランダムドットという「意味をもたない情報の中から生成された意味」を利用する、というアプローチに基づいた動画 CAPTCHA である。この CAPTCHA では、ランダムドットパターンの背景の上で、やはりランダムドットパターンが描かれた「正解パターン」を 1 フレーム毎に移動させながら表示する (図 2)。回答者はこの動画の中で「正解パターン」の領域の座標を入力し、それが正しい座標であれば人間とみなされる。人間であれば、複数のランダムドットを一つのまとまりとして認識可能であるので、ランダムドットパターンの中から正解パターンを認識・選択

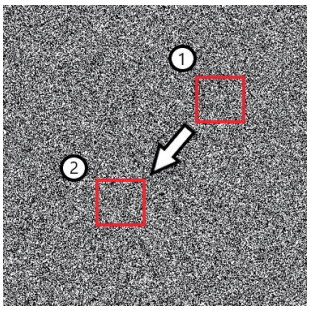

図 2. SNOW NOISE CAPTCHA のコンセプト図

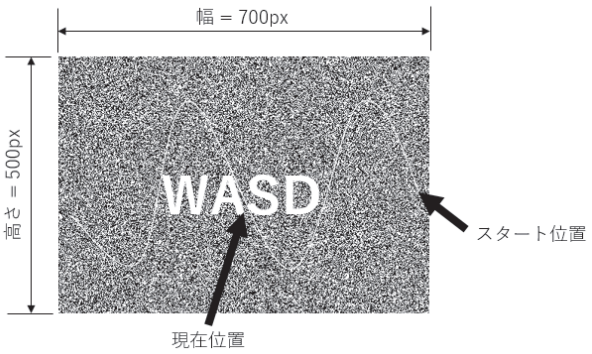

図 3. 提案 CAPTCHA のコンセプト図

可能である。一方ボットは、1 フレーム毎の静止画像から回答を探すこととなり、正解パターンの発見は困難である。これを利用して人間とボットの区別が可能となる。

ただし、SNOW NOISE CAPTCHA では、正解パターンの領域がウィンドウ中に必ず存在するため、総当たり攻撃を受けると偶然突破されてしまう懸念がある。また、動画から数フレームを抽出し、それらの差分を利用して正解パターンの位置や形を把握する攻撃も考えられる。

3. 提案手法

3.1 目的と提案 CAPTCHA

本研究では、ランダムドットパターンの性質を活用し、プログラムによる自動化攻撃に耐性のある CAPTCHA を提案する。提案する CAPTCHA は文字列 CAPTCHA をベースとしたものであり、ランダムドットパターンの背景の上を、別のランダムドットパターンで描かれた文字列を 1 フレーム毎に移動させながら表示する。この文字列を判読できるか否かで回答者が人間かボットかを判断するものである。

図 3 は、作成した CAPTCHA から 1 フレームを抜き出したものである。表示される文字列は、図中のスタート位置からウィンドウ内に進入し、白線に沿って移動する（実際の画面には、白線は表示されない）。この図では、現在位置として示された位置に文字列が表示されているが、実際にランダムドットパターンで表示すると識別できないので、白抜きの文字で表示している。

3.2 物体追跡技術への対処

提案する CAPTCHA を自動プログラムで解答する方法として、2 フレームを抽出し、その差分を利用することで表示されている文字列を発見する方法や、オプティカルフロー（図 4）と呼ばれる、画素単位の物体の動きをベクトルで表し、表

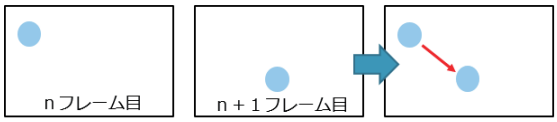

図 4. オプティカルフローの例

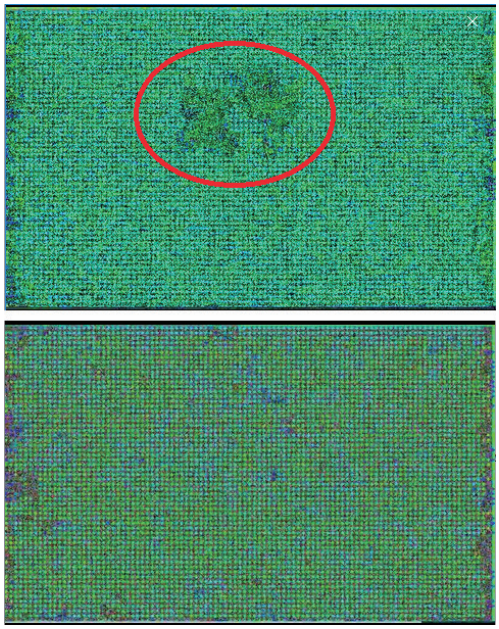

図 5. オプティカルフローへの耐性検証

示される文字列の動きを追跡することで文字の形を抽出する方法が考えられる。これらの方法により抽出した文字の形に OCR を適用することで、自動プログラムによる文字の判読が可能となる。

そこで提案手法では、まず、文字列だけでなく、背景も文字列とは異なる方向に移動させることとした。なぜなら、ランダムドットパターンの背景上が静止しており、文字列だけを動いていると、フレームの差分を利用することによって、回答すべき文字列が容易に判読可能となるからある。

次に、オプティカルフローを利用した攻撃への対策として、背景と文字列の上に、透過度を持つランダムドット前景（透過度 0.4）を 2 枚表示し、文字列のランダムドットパターンの動きをカモフラージュさせることにより、正解文字列の難読化を図った。

妨害用の前景を加える前と加えた後の CAPTCHA に対して、実際にオプティカルフローを適用したものが図 5 である。図 5 上段では前景を加えていないのでオプティカルフローによって文字の位置が分かってしまい、文字を判読されてしまう可能性がある。一方、図 5 下段では前景を加えたことにより、文字の位置が分からなくなっている。

以上の結果より、提案 CAPTCHA はオプティカルフローを用いた攻撃に対しての耐性が示された。

4. 評価実験

4.1 実験目的

今回は、提案 CAPTCHA が人間のユーザーにとって解答可能なものであることの確認と、ユーザビリティ評価による CATPCHA としての実用性について調べる。

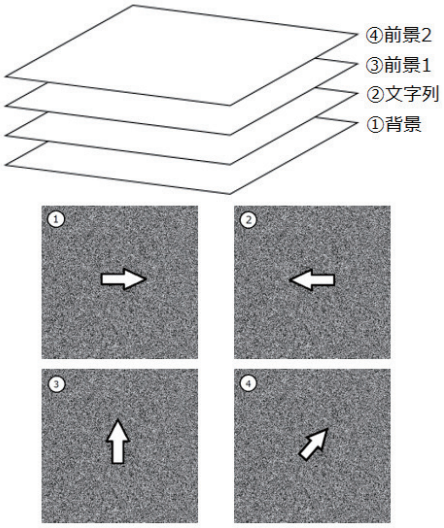

図 6. レイヤー構造と各レイヤーでのランダムドットの動き

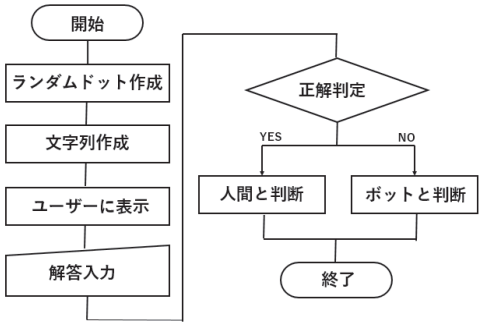

図 7. 認証手順のフローチャート

4.2 実験用システムの実装

提案 CAPTCHA のユーザビリティ評価のための実験環境を JavaScript で実装した。背景や正解文字列を満たすランダムドットパターンを構成するドットの色は、明色時 (R, G, B) = (255, 255, 255)、暗色時 (R, G, B) = (0, 0, 0) の 2 色のどちらかが無作為に選ばれる。実験環境では、提案 CAPTCHA の実現に、HTML5 の canvas 要素を利用している。背景となるランダムドットのレイヤーを最背面として、正解文字列のレイヤー、妨害用の透過したランダムドットのレイヤー 2 枚の計 4 枚を重ねて描画する (図 6)。また各レイヤーは、それぞれ異なる方向に移動するようにした。

システムの動作の流れを図 7 に示す。CAPTCHA プログラムが動作を始めると、背景用のランダムドットパターン、表示される文字列用のランダムドットパターン、妨害用の 2 つのランダムドットパターンの計 4 つが生成され、それぞれが互いに異なる方向に移動しながら表示される。ランダムドットパターンは、毎回全て新たに自動生成される。表示される文字列も、毎回異なったものとなるように、A〜Z の 26 文字からランダムに文字を選択して文字列を構成する。

ユーザーは表示されている文字列を指定のテキストボックスに入力し、回答用のボタンをクリックすると、それがサーバに送られる。それに基づき、ユーザーが入力した文字列と表示されている文字列が完全に一致しているか否かの判定が行われる。

表 1. 実験環境

OS	Windows 10(64bit)
CPU	Intel(R) Core(TM) i7-4770S CPU @ 3.10GHz
メモリ	8GB
解像度	1,920 × 1,200

4.3 実験方法

実験の被験者は宮崎大学に所属する学生 20 名である。

実験は、以下の手順で行なった。まず被験者には、提案 CAPTCHA を解いてもらう前に一通りの操作手順を説明し、その後、10 回の解答を行ってもらった。被験者は、出題動画中に表示される文字列を判読し、指定のテキストボックスに入力し、解答送信ボタンをクリックする。その際、被験者が入力した回答の正否、および、ページ表示から解答までの所要時間を記録した。

実験で使用した計算機環境を表 1 に示す。

各被験者の正答率や所要時間の記録を完了した後、アンケート調査により、ユーザビリティ評価を行った。ユーザビリティの数値的な評価を実現するため、ここでは、SUS (System Usability Scale)[7] を用いた。またこのアンケートでは、現在の主流である文字列 CAPTCHA と提案 CAPTCHA の比較も行った。

SUS は、ユーザビリティについての主観的な評価の指標であり、John Brooke が 1986 年に開発したものである。当初は、当時の文字ベースの PC の評価に使われていたが、その後、携帯電話やハードウェア、IVR などの評価軸としても利用されてきている。

以下に、本研究用に変換した 10 項目のアンケート内容を示す。

1. この CAPTCHA をしばしば利用したいと思う。
2. この CAPTCHA を利用するには説明が必要となるほど複雑であると感じた。
3. この CAPTCHA は容易に使いこなす事ができると思った。
4. この CAPTCHA を利用するのに専門家のサポートが必要だと感じる。
5. この CAPTCHA にあるコンテンツやナビゲーションは十分に統一感があると感じた。
6. この CAPTCHA では一貫性のないところが多々あったと感じた。
7. たいていの人は、この CAPTCHA の利用方法をすぐに理解すると思う。
8. この CAPTCHA はとても操作しづらいと感じた。
9. この CAPTCHA を利用できる自信がある。
10. この CAPTCHA を利用し始める前に知っておくべきことが多くあると思う。

SUS の集計方法は、次の様に行う。まず、すべての項目は 0 から 4 の 5 段階で評価する。その上で、奇数番目の質問項目のスコアは回答番号から 1 を引いたもの、偶数番目の質問項目のスコアは 5 から回答番号を引いたものとし、その結果を足しあわせた合計数値を 2.5 倍たのち、0 から 100 のスケールへ変換する。

表 2. 実験・アンケート結果

平均正答率 [%]	97.5
平均所要時間 [秒]	8.06
最小所要時間 [秒]	18.0
最大所要時間 [秒]	3.7
平均 SUS スコア	89

各項目のスコアを N_1 から N_{10} とすると、合計スコア S は式 (1) で表すことができる。

$$S = (\sum_{i=1}^{10} N_i) \times 2.5 \tag{1}$$

スケール後の数値が高いほど、システムとして良い評価が与えられる。SUS スコアは、Jeff Sauro らによる調査結果 [7] から平均スコアが 68 とされており、ユーザビリティに優れた上位 10%に入るには、80.3 を超えるスコアが必要とされている。

4.4 実験結果

被験者 20 名が提案 CAPTCHA を 10 回ずつ解いて得られた 200 件のデータから、このデータ群の平均正答率、平均所要時間、最大所要時間、最小所要時間を算出した。これらの値と、SUS によるユーザビリティ評価の平均スコアを表 2 に示す。

表 2 に示す様に、被験者 20 名の平均正答率は、97.5%、解答における平均所要時間は、8.06 秒であった。

被験者 20 名、計 200 回の解答に対して、不正解と判定されたは 5 回であった。その主な理由は、表示された文字を、それに似た文字に読み間違えたものであった。

4.5 考察

実験結果から、平均正答率は 97.5%であり、高い水準となっている。所要時間においては、一般的な文字列 CAPTCHA の平均所要時間は約 12 秒であるのに対して、提案 CAPTCHA は約 8.06 秒であり、文字列 CAPTCHA より若干短い時間で解ける CAPTCHA となっていると考えられる。ただし、今回の実験は、一般的な文字列 CAPTCHA よりも少ない文字数で実施したため、提案 CAPTCHA での文字列識別に要する時間が、文字を歪ませる形式の文字列 CAPTCHA よりも短くて済む、と判断することはできない。

次に、表 2 より、今回の SUS に基づいたアンケート調査のスコアは 89 であり、優れたユーザビリティを示すスコア 80.3 を超える結果となった。また、提案 CAPTCHA と一般的な文字列 CAPTCHA のどちらを使いたいか、アンケートで尋ねたところ、1 名を除き、提案 CAPTCHA を選択していた。これらのことから、 提案 CAPTCHA は実用的であるといえる。

その一方、被験者のコメントとして、似たような文字が判別しづらいことや、見続けると目が疲れるといったことが挙げられた。そのため今後は、文字の大きさやフォントなど、人間にとって見やすい表示方法を検討する必要がある。

5. まとめ

本研究では、人間の高度な認識能力に着目し、自動プログラムによる攻撃への耐性を持たせた CAPTCHA を提案した。また、提案方式の CAPTCHA を実装し、実用性とユーザビリティ評価を行う実験を行った。実験の結果、人間であるユーザーが解いた場合は高い正答率を示し、ユーザビリティ評価の指標となる SUS スコアも高い数値となった。これらのことから、提案 CAPTCHA が実用的であることが確認できた。

今後は、今回確認することができなかった新たな攻撃方法による自動プログラムへの耐性を検証しつつ、ユーザビリティ面での問題点の解決に向けて検討していきたい。

参考文献

1) L. von Ahn, M. Blum, N. Hopper, and J. Langford: CAPTCHA: Telling humans and computers apart, Advances in Cryptology, Eurocrypt'03, vol.2656 of Lect. Notes Comput. Sci., pp.294-311, 2003.
2) Greg Mori, Jitendra Malik: Recognizing Objects in Adversarial Clutter:Breaking a Visual CAPTCHA, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), Vol.1, p.134, 2003.
3) Gabriel Moy, Nathan Jones, Curt Harkless, and Randall Potter: Distorition Estimation Techniques in Solving Visual CAPTCHAs, proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), 2004.
4) 渡邊恵太: Texture World, 不思議アート, http://fushigi-art.com/wordpress/?p=242, (2018/03/06 閲覧).
5) Tom Stafford and Matt Webb: Mind Hacks, Oreilly & Associates Inc, 2004.
6) 藤田真浩、池谷勇樹、米山可児、西垣正勝: SNOW NOISE CAPTCHA: 無意味な情報を利用した動画 CAPTCHA の提案, 情報処理学会研究報告 (CSEC), Vol.29, pp.1-7, 2014.
7) Jeff Sauro: MEASURING USABILITY WITH THE SYSTEM USABILITY SCALE (SUS), Measuring U, https://measuringu.com/sus/, (2018/02/01 閲覧).