情報処理学会論文誌 Vol.62 No.12 1864–1878 (Dec. 2021)

リレーアタック耐性とボット耐性の両立を目指したインタラクティブな動画CAPTCHAの提案と評価

油田 健太郎[1,a)] 臼崎 翔太郎[1] 山場 久昭[1] 片山 徹郎[1] 椋木 雅之[1] 朴 美娘[2] 岡崎 直宣[1]

受付日 2021年3月8日, 採録日 2021年9月9日

概要：Webサービスへの不正対策としてCAPTCHAが用いられている．CAPTCHAは，人間には容易に解答できるがコンピュータ（ボット）には困難な問題を出題し，正しい解答をした者を人間と判断するシステムである．CAPTCHAはリレーアタックと呼ばれるインターネット上の報酬に誘引された人間を利用してCAPTCHAを解読させる方法に対して脆弱である．そこで，本論文ではリレーアタックを行った場合に生じる通信遅延に着目し，リレーアタックでのCAPTCHAの解答を困難にすることを目指したCAPTCHA方式を提案する．提案手法は，連続的に移動するオブジェクトをマウスカーソルで追跡し，その追跡が成功していた時間が所与の値を超えているか否かで人間かボットかを判別する．提案手法を評価するために，リレーアタックの実験環境を構築して実験を行った．次に，自動プログラムを用いた攻撃に対しても評価を行った．その結果，提案手法はリレーアタックと自動プログラムを用いた攻撃に対してロバストな方式であることを明らかにした．

キーワード：CAPTCHA，ボット，リレーアタック，自動攻撃

Proposal and Evaluation of the Interactive Video CAPTCHA for Resistant to both Relay Attack and Automated Attack

Kentaro Aburada[1,a)] Shotaro Usuzaki[1] Hisaaki Yamaba[1] Tetsuro Katayama[1] Masayuki Mukunoki[1] Mirang Park[2] Naonobu Okazaki[1]

Received: March 8, 2021, Accepted: September 9, 2021

Abstract: CAPTCHA can be used to protect Web services against unauthorized access. CAPTCHA is designed to detect automated programs (called bots) by requiring them to perform tasks that are easy for humans but difficult for automations. CAPTCHAs are vulnerable to relay attacks in which the challenges are relayed to remote human-solvers who hope for a reward. Our new CAPTCHA uses delay time between communications to prevent relay attacks. In our CAPTCHA, users have to recognize the target object from a number of randomly appearing decoy objects and tracks it with his/her mouse cursor. To pass the test, the user must track the target for a given amount of time. We constructed an experimental environment in which relay attack can be simulated, made a series of experiments in order to evaluate the performance of the proposed method. Next, we implemented an automated attack for applying to our CAPTCHA and evaluate its resistance to automated attacks. Our results showed the robustness of our proposed method against relay attack and automated attacks.

Keywords: CAPTCHA, bot, relay attack, automated attack

1. はじめに

Webサービスが普及した今日のネットワーク社会において，CAPTCHA（Completely Automated Public Turing test to tell Computers and Humans Apart）は重要な役割を負うようになってきている．Webサービスの普及によ

[1] 宮崎大学
University of Miyazaki, Miyazaki 889–2192, Japan
[2] 神奈川工科大学
Kanagawa Institute of Technology, Atsugi, Kanagawa 243–0292, Japan
[a)] aburada@cs.miyazaki-u.ac.jp

り，誰でも様々なサービスを利用することが可能となっているが，それらの Web サービスに対して，ボットと呼ばれる自動プログラムを用いた不正行為が行われている．たとえば，ボットを用いてメールサービスのアカウントを大量取得し，スパムメールの送信に利用するなどの事例があげられる．このような不正行為を防止するために広く利用されているのが CAPTCHA と呼ばれる反転チューリングテストである [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困難な問題を出題し，正しい解答をした者を人間と判断する．

これまで，様々な CAPTCHA が研究されてきているが，それを破るためのプログラムもまた日々工夫されてきている．初期の CATPCHA には，Web ページ上に歪みやノイズを加えた文字列画像を提示し，Web サイトの閲覧者がその文字列を判読できるか否かを試す文字列 CAPTCHA がある．しかし，OCR 技術の進歩や解読アルゴリズムの向上により，文字列 CAPTCHA は容易に突破されるようになってきている．そのため，動物や物などの画像を識別する人間の高度な能力を利用する画像 CAPTCHA や文字列 CAPTCHA を動画へ応用した動画 CAPTCHA など数多くの方式が提案されてきた．

ところがこれらの CAPTCHA を回避する手法として，それを破るプログラムを開発するというアプローチとは異なる視点を持つ，リレーアタックと呼ばれる攻撃手法が用いられることがある [2]．リレーアタックは，インターネット上の一般ユーザや報酬に誘引された人間（以下，幇助ユーザと呼ぶ）を利用して CAPTCHA を解読させ，その解答を利用する手法である．リレーアタックでは CAPTCHA を解くのが人間であるので，コンピュータを想定した対策では効果がなく，新たな対策が求められている．

そこで本研究では，リレーアタックへの耐性を持つ CAPTCHA を提案するともに，ボット耐性が損なわれないような工夫を加えて，その性能を評価した結果を報告する．提案する CAPTCHA は，リレーアタックを行った際に生じる遅延時間に着目し，リレーアタックでの CAPTCHA の解答を困難にすることを目指している．提案方式は，リアルタイムのマウス操作を必要とするインタラクティブ性を利用する．具体的には，ランダムな位置に出現する複数の妨害オブジェクトの中から，連続的に移動してその位置を変化させる移動オブジェクトを発見・特定し，マウスカーソルで追跡できるか否かで人間かボットかを判別する．リレーアタックでは，攻撃者が幇助ユーザに CAPTCHA の出題画像を転送する通信の遅延時間が発生するため，提案方式 CAPTCHA の場合，攻撃者に提示されている動画と幇助ユーザに中継されている動画には，時間のずれが生じ，リレーアタックによる移動オブジェクトの追跡が困難になると考えられる．

本論文では，リレーアタックを模擬試行（再現）し，提案方式 CAPTCHA への解答が，CAPTCHA の転送で生じる遅延時間のために困難になることを検証し，提案 CAPTCHA がリレーアタック耐性を持つことを確認する．一方，画像処理に基づいた自動的な攻撃と機械学習に基づいた自動的な攻撃として，それぞれ mean shift 法によるものと Deep Learning 技術よるものの 2 通り実装し，それらを使用して提案 CAPTCHA がボットへの耐性を持つことを確認する．

以下，本論文では 2 章で既存の CAPTCHA の分類とリレーアタックについて述べる．3 章では，提案する CAPTCHA の設計，リレーアタック耐性やボット耐性について述べる．4 章では，リレーアタック耐性の検証実験について，5 章では，ボット耐性と追跡精度について，考察を述べる．6 章では，Deep Learning 技術を適用した攻撃手法による実験について述べる．7 章でまとめと今後の課題について述べる．

2. 関連研究

2.1 CAPTCHA の分類

2.1.1 文字列 CAPTCHA

文字列 CAPTCHA は，最も広く利用されてきた CAPTCHA 方式であり，人間には認識できるが，ボットには認識することが困難な歪みやノイズを含んだ文字を出題する（図 1）．

文字列 CAPTCHA 方式のメリットは，実装が容易である点と，Web システムへの導入が簡単である点，総あたり攻撃に強い耐性がある点である．一般的な文字列 CAPTCHA は，英字 52 字（大文字と小文字を含む）と数字 10 字の合計 62 字の英数字が用いられるので，CAPTCHA の文字数が a だとすると，文字列の画像のパターン数は 62^a 通りである．すなわちボットがこの文字列 CAPTCHA を総あたりで突破する場合，最悪で 62^a 通りの答えを試さなければならない．

文字列 CAPTCHA のデメリットは，OCR（光学文字認識）攻撃への耐性が弱いことである．技術の発達にともない，OCR の文字列の認識精度が向上し，難読化を施した文字であっても，ボットによって突破されてしまう事態が発生するようになった [3], [4]．これに対処しようと，文字列に加える変形やノイズを大きくすることでボットへの耐性を向上させようとしても，そのような文字は，人間にとっ

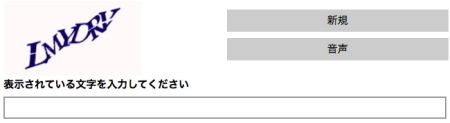

図 1 Microsoft 社のサイトで利用されている CAPTCHA [*1]
Fig. 1 Microsoft's CAPTCHA.

[*1] マイクロソフト社の Windows Live サービスより転載

図 2 Asirra *2
Fig. 2 Asirra.

ても判読が難しくなってしまい，人間の正答率まで低下させてしまう．

2.1.2 画像 CAPTCHA

文字列 CAPTCHA 方式の弱点を解消する手法として，画像 CAPTCHA が提案されている．画像 CAPTCHA 方式は，具体物の画像を用いることで，人間とコンピュータを判別する．出題する問題の種類は様々あり，用いる画像の枚数や解答方式に違いがある．

画像 CAPTCHA 方式のメリットは，文字列 CAPTCHA 方式の脅威であった OCR 機能を持ったボットが通用しないことや，人間の直感的な画像認識を用いるため求解の負荷が低いこと，Asirra [5]（**図 2**）では画像を選択するだけで良いことに代表されるように解答方法が容易である点があげられる．

一方そのデメリットは，ボットが誤って CAPTCHA の判定テストを通過してしまう確率 False Acceptance Rate（FAR）が高い点である．FAR を下げるためには，出題する画像の選択肢を増やす方法が考えられるが，大きな表示スペースが必要になったり，各画像が小さくなり，使い勝手が悪くなったりしてしまう．

2.1.3 動画 CAPTCHA

動画 CAPTCHA 方式は，文字列 CAPTCHA や画像 CAPTCHA の拡張方式となっており，静的な画像の出題形式の後継として開発された．

動画 CAPTCHA のメリットとして，動画を用いることにより，文字列の歪みやノイズなどの従来の文字列 CAPTCHA の難読化に新しい要素を追加できることである．このような，難読化のバリエーションの増加は，過度な歪みやノイズによって，人間にも文字列 CAPTCHA が読めなくなる事態を抑制できることが期待できる．

2.2 リレーアタック

2.2.1 リレーアタックの概要

典型的なリレーアタックは，(1) 攻撃者が CAPTCHA を提示する Web ページから CAPTCHA の出題画像を取得し，(2) 幇助ユーザに転送，(3) 報酬を与えることと引き換えに CAPTCHA の解読を行ってもらい，(4) その解答を利用することで CAPTCHA を突破する手法である（**図 3**）．

*2 文献 [5]，p.1 の図 1 より転載

図 3 リレーアタックの一例
Fig. 3 An example of relay attack.

図 4 DCG-CAPTCHA *3
Fig. 4 DCG-CAPTCHA.

問題画像の取得や幇助ユーザへの問題の転送は，攻撃者の作成したプログラムで自動的に行われる．リレーアタックでは，CAPTCHA の解読を行うのが人間であるため，この攻撃手法を実行されると，高い確率で CAPTCHA を突破されてしまう．

2.2.2 リレーアタックに対応する既存研究

DCG-CAPTCHA は，簡単なミニゲーム形式の CAPTCHA である．この CAPTCHA は，指示に適するオブジェクトをマウスなどで選択させ，その選択が正しければ人間と見なすものである [6]．**図 4** は，その一例であり複数の異なる形状のオブジェクトの中から，青いエリアのオブジェクトと同じ形状のものを選択し，青いエリアのその形状のオブジェクトの位置にドラッグ&ドロップで配置できれば，ユーザを人間と見なす．また，DCG-CAPTCHA は常にオブジェクトが移動する動的な CAPTCHA である．右の白いエリアの図形はこのエリア内を移動しつづけている．

この CAPTCHA をリレーアタックで突破するには，CAPTCHA のフレーム画像を幇助ユーザに送信し続けなければならない．このとき，幇助ユーザが目視している DCG-CAPTCHA は，通信の遅延などにより中継元に表示されているものとずれが生じる．DCG-CAPTCHA を解く場合，オブジェクトの移動などにリアルタイムで対応しなければならない．そのため，幇助ユーザの解答を利用したとしても，生じる通信の遅延により，リレーアタックでの解答が困難になる．この点に着目し，ユーザと CAPTCHA とのインタラクションのタイミングを検査することでリレーアタックの検出が実現されている [7]．

しかし，同形状のオブジェクトを認識することや移動オブジェクトのフレーム画像を解析してプログラムで追跡することは容易にできるため，ボットによる攻撃への耐性は低いといえる [8]．

*3 文献 [6]，p.2 の図 1 より転載

3. 提案手法

3.1 想定するリレーアタック

リレーアタックの成功報酬は 1,000 問解いて 1US$程度と単価が低いうえ [9]，提案する CAPTCHA を 1 問解くのには 10 秒程度要するため，時間あたりの報酬額が低い．このことから，幇助ユーザは，国民の平均的な所得額が低い新興国から参加していることが多いと考えられる．そこで本研究では，CAPTCHA サーバと中継 PC 間は低遅延である（同一国内にあるまたは高品質な回線状況である）が中継 PC と幇助ユーザ間には大きな遅延があることを想定する．また，リレーアタックを実施するために専用のソフトウェアを用意するようなコストはかけず，攻撃者が用意した PC を，ネットワークを介して遠隔地のコンピュータを操作するソフトウェア，たとえば Virtual Network Computing（VNC）を用いて，補助ユーザに操作させて指定した CAPTCHA を解かせる，というアプローチをとっているものとする．

3.2 提案手法の概要

リレーアタックとボット耐性を両立させるにあたり，(1) リアルタイムにインタラクティブな操作を要求すること，(2) その操作の対象物が視覚的特徴で識別できないように難読化することの 2 つを同時に備えた CAPTCHA を設計する方針をとった．具体的には以下のようにした．

(1) 連続的に移動するオブジェクト（以降，移動オブジェクトと呼ぶ）をマウスカーソルで追跡する方法を採用する．このとき，移動オブジェクトはランダムなタイミングでランダムに方向を変える．移動する物体をマウスカーソルで追跡するような作業は，通信時間の遅れのために困難であり，リレーアタックの阻止に有効であると期待できる（詳細は 3.4 節を参照）．

(2) 移動オブジェクトの形状の工夫と，移動オブジェクトと同じ形状の図形（妨害オブジェクトと呼ぶ）の導入により，ボットによる移動オブジェクトの探知を阻害する．前者については 3.5.1 項で，後者については 3.5.2 項で説明する．

また，提案する CAPTCHA のように，動画を利用したインタラクティブな処理を利用する場合，CAPTCHA サーバと PC 間の通信に対する，以下のような攻撃方法も考えられる．

(1) CAPTCHA プログラムの解析
(2) サーバ・クライアントユーザ間のプロトコル解析による正解の自動算出

そこで，これらの方法に対しても十分な耐性が得られるように，動画 CAPTCHA を前提としたうえで動画フレームを直接送るアプローチをとることとする．

提案する CAPTCHA 方式 [10], [11] については，リレーアタックへの耐性の評価 [10] とボットへの耐性の評価 [11] を行い，それぞれ速報（letter）として発表している[*4]．本論文では，提案方式の設計について詳細を述べるとともに，新たに Deep Learning 技術を適用した攻撃手法よる実験を追加し，提案手法の有効性を示す．

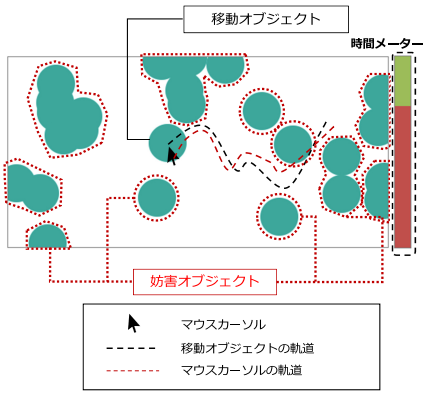

図 5 提案方式の CAPTCHA
Fig. 5 CAPTCHA's concept.

3.3 提案する CAPTCHA の解答

提案する CAPTCHA（**図 5**）は，連続的に移動する移動オブジェクトをマウスカーソルで追跡できるかどうかで直接アクセスしてきている人間（以降，正規の人間と呼ぶ）であるか，リレーアタックであるかを判別する．提案する CAPTCHA が起動すると，まず START ボタンが表示され，これをクリックすると，図 5 のような画面が表示される．移動オブジェクトはただちに移動を開始する．ユーザは移動オブジェクトを認識してマウスカーソルで追跡を試みる．なお，移動オブジェクトの初期位置はランダムに選ばれる．

正規の人間であると判断する基準は，ある制限時間中に，追跡に成功していた時間の長さが，別途指定する閾値以上であるときとする．ただし「追跡に成功している」とは，マウスカーソルの座標が，移動オブジェクトの図形内部にあるときのことである．また，移動図形の中にあったマウスカーソルが一度図形の外部に出てしまっても，追跡時間はリセットされない．すなわち，追跡に成功している時間の総和が閾値を超えれば良いものとする．

閾値の値は，ボットとリレーアタックによる攻撃で達成することが難しい値に設定すべきであり，後述する実験で検討を行う．この実験では，制限時間は，マウスカーソルが最初に移動オブジェクトの図形の内部に入ってから 10 秒としている．また，移動オブジェクト（および妨害オブジェクト）の形状は真円としており，移動オブジェクトの座標 (O_x, O_y) とマウスカーソルの座標 (M_x, M_y) から求まる距離 d が，オブジェクトの半径 r より小さい場合に追跡できていると見なす（**図 6**）．

[*4] 文献 [10], [11] は OpenAccess である．また，当該の文献の PDF ファイルには，提案手法の動画を埋め込んでいる．

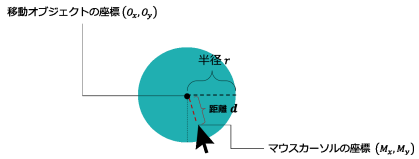

図 6 追跡の判定
Fig. 6 Judgement of tracking.

$$d = \sqrt{(O_x - M_x)^2 + (O_y - M_y)^2} \quad (1)$$

移動オブジェクトの動きとしては，速さを一定としたうえで，現在の位置とは異なるランダムな位置を目的地に選び，この 2 点間の長さが別途与える値になるようなベジエ曲線を描き，移動オブジェクトはこの曲線上を移動して目的地に到達する．目的地に到達すると，新たな目的地を選択し，この操作を繰り返していく．よって，選ばれた目的地が現在位置から離れていれば，目的地までの経路は直線に近くなり，短ければ，大きく湾曲した経路となる．

実験にあたっては，提案する CAPTCHA を実装した図 5 のシステムを利用した．オブジェクト表示領域の右に，ユーザが時間を直感的に把握できるようにするための時間メータを配置している．このメータは，ユーザが移動オブジェクトの追跡を開始してからの経過時間や追跡成功時間がリアルタイムに反映される．灰色の帯全体が制限時間（後述の実験では 10 秒）の長さを表し，解答の経過時間に対応する長さが下から緑色に変わっていく．さらに，移動オブジェクトの追跡に成功している時間に対応する長さの分が赤色になる．

3.4 リレーアタックへの耐性

提案する CAPTCHA により，リレーアタックが困難であることについて説明する．提案方式では，CAPTCHA サーバから正規のユーザや中継 PC に対して，動画のフレームを送り続け，それらのユーザからマウスカーソルの座標を受け取る．これをリレーアタックで突破するには，CAPTCHA のフレーム画像を幇助ユーザに送信し，幇助ユーザから解答情報（マウスカーソルの座標）を得ることが必要である．

図 7 に提案 CAPTCHA に対してリレーアタックを行ったときの通信についてのシーケンス図を示す．図 7 で用いている記号の意味を以下に示す．

Ox_t, Oy_t：時刻 t における中継 PC 上の移動オブジェクトの座標．

Mx_t, My_t：時刻 t における幇助ユーザの PC 上の移動オブジェクトの座標．

Δt_1：中継 PC から幇助ユーザに CAPCHA のフレーム画像が送信されてくるまでの時間．

Δt_2：幇助ユーザから中継 PC に解答に用いるマウスカーソルの座標が送信されてくるまでの時間． □

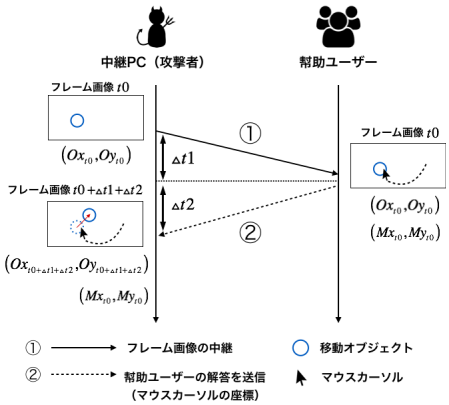

図 7 リレーアタックのシーケンス図
Fig. 7 Sequence diagram of relay attack.

幇助ユーザには，移動オブジェクトが時刻 t にはが幇助ユーザの PC 上で座標 (Mx_t, My_t) にいるように見えるので，そこにマウスカーソルを移動させようとする．しかし中継 PC 上では，移動オブジェクトは座標 (Ox_t, Oy_t) に移動してしまっており，さらに，中継 PC 上でマウスカーソルが座標 (Mx_t, My_t) に移動するのは時間 Δt_2 後になるので，移動オブジェクトはその時点ではさらに違った位置に移動してしまっている．マウスカーソルを移動オブジェクトの現在位置ではなく，この遅れ時間を考慮した予想到達位置に移動させようとする解法が考えられるが，提案 CAPTCHA において，移動オブジェクトはランダムな方向へ連続的に移動するため，移動オブジェクトの位置を予想することは困難である．

またこの環境でマウスを操作するのは，手元の PC でマウスを利用するように容易ではない．手元の PC を操作しているとき，マウスの移動に応じて即座にカーソルの位置が変化するので，画面上のカーソルの現在位置と移動させたい場所をフィードバックして，カーソルを任意の位置に持っていくことができる．しかし，幇助ユーザが座標 (Mx_t, My_t) に向けてマウスを動かしても，その結果としてのマウスカーソルの移動が幇助ユーザの PC の画面に反映されるは，時間 $\Delta t_2 + \Delta t_1$ 後になる．マウスカーソルの現在位置が座標 (Mx_t, My_t) とのずれが判明するまでにこれだけの時間遅れがあり，しかも，移動オブジェクトは刻々とその位置を変えているため，その追跡は困難である．

3.5 ボットへの耐性

ボットを利用した攻撃方法としては，(1) テンプレートマッチング，(2) AND 演算，(3) 機械学習を用いる方法，(4) mean shift 法などが考えられる．このうち，提案手法はテンプレートマッチングや AND 演算では移動オブジェクトを追えないように設計しているのでそれらについて説明する．また (3) については Deep Learning を対象に考察を行う．以下，(1) について 3.5.1 項，(2) について 3.5.2 項，(3) について 3.5.3 項，(4) について 3.5.4 項が対応する．

3.5.1 妨害オブジェクトの導入

提案手法では，ボットによる物体認識を困難にするために，移動オブジェクトと同形状，同色，同じ大きさの妨害用のオブジェクトを複数配置する（図 5）．移動オブジェクトを自動的に追跡するためには，フレーム画像中からリアルタイムで移動オブジェクトを検出する必要がある．ボットが自動的に移動オブジェクトを追跡しようとする場合，フレーム画像を解析して追跡対象のオブジェクトを見つけようとするが，提案手法のフレーム画像には，同じ形・色・大きさのオブジェクトが散らばっているようにしか見えないため，移動オブジェクトを特定することは難しくなる．逆に人間にとっては，フレーム画像に同じオブジェクトが散らばっているだけだとしても，動画として見たときには移動オブジェクトを見つけることは容易にできるため，CAPTCHA として成立する．

提案する CAPTCHA 方式では，妨害オブジェクトと移動オブジェクトとの間に視覚的特徴の違いがないため，追跡対象のパターンを用いて，フレーム画像中から移動オブジェクトを検出することは困難である．たとえば，テンプレートマッチングのような追跡対象のパターンを用いる手法では，移動オブジェクトを追跡することは困難である [10]．

3.5.2 オブジェクトの形状の工夫

提案する CAPTCHA では，移動オブジェクト（および妨害オブジェクト）を破線の円とすることにより，AND 演算による検出を困難としている．

AND 演算では，2 値化した 2 枚のフレームから重なった部分（共通領域）を抽出できる．提案方式において，移動オブジェクトの移動は連続的であるため，連続したフレーム間での位置の変化が小さい，一方，妨害オブジェクトは 1 フレームごとにランダムな位置に出現するため，連続したフレーム間での位置の変化が大きい．そのため，図 5 に示したような塗りつぶした円の場合，連続したフレームに AND 演算をすると，位置の変化が小さい移動オブジェクトは共通領域が大きく抽出されてしまい，逆に，妨害オブジェクトは一般に抽出される共通領域が少なくなり，除外することができてしまう（**図 9**）．

そこで提案手法では，ボット対策としてオブジェクトを**図 8** に示すような，塗りつぶしのない，破線の円としている（図 5 ではオブジェクトを塗りつぶして円として描いているが，これは本手法の考え方の説明のためである）．各構成要素は，点が小さく破線になっているため，共通領域が発生しにくく，AND 演算で移動オブジェクトの位置を推測することが困難になる（**図 10**）．

また，図 8 のオブジェクトは，1 フレームごとにオブジェクトの構成要素が位置を変えるが，人間には回転しているように見えるため，視認性が低くなりにくい．

図 8 に説明した 8 つの点により円状のオブジェクトが構成されることを攻撃者が知っている場合，T フレームにおいて 2 種類のテンプレートを用意してテンプレートマッチングにより検出し，T+1 フレームではその T フレームで見つかったものの近くのみを探すことで可能になると考えられる．これについては，パターン数を増やすことで対応が容易に可能であるが，ユーザビリティにどのような影響があるかをあわせて調査する必要がある．

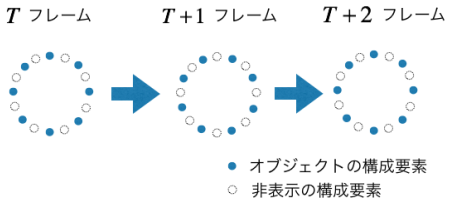

図 8 破線の輪郭線によるオブジェクト
Fig. 8 an object drawn as a dotted outline.

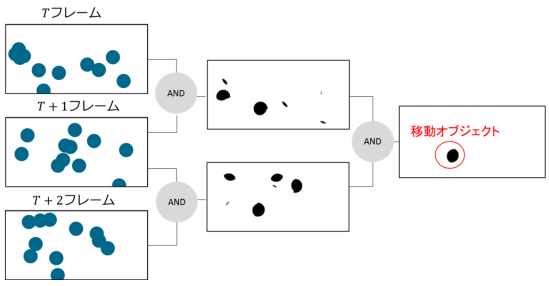

図 9 AND 演算による移動オブジェクトの抽出
Fig. 9 Extraction of the target object by AND operation.

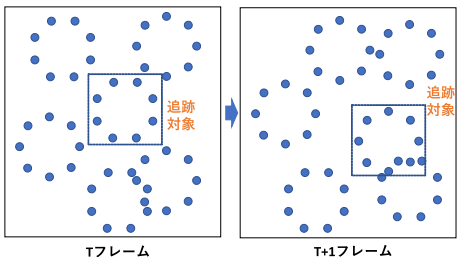

図 10 追跡対象の移動
Fig. 10 Movement of the moving object.

3.5.3 Deep Learning 技術に対する耐性

次に，機械学習の一例である Deep Learning 技術に対する耐性について考察する．Deep Learning による画像認識においては，一般的に教師あり学習が用いられ，たとえば，膨大な数の犬や猫の画像を学習することで，未知の画像についても犬か猫かを判断できるようになる．本提案手法は，移動オブジェクトと妨害オブジェクトは同じ丸型の形状である．また，移動オブジェクトは図 8 に示したとおりフレームごとに形を変えるが，それを含めても 2 パターンしかない．よって，パターン数が少ないため，学習に必要な負荷を考えると単に物体を認識する目的で教師あり学習を利用することはあまり現実的ではないと考えられる．

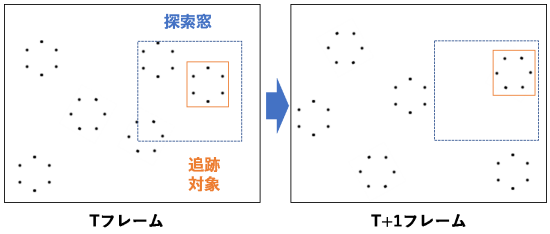

図 11 mean shift 法による追跡
Fig. 11 Tracking with mean shift.

3.5.4 mean shift 法についての検討

最後に，mean shift 法 [12], [13] による物体追跡について検討する．mean shift 法は，与えられた初期位置の近傍で，密度関数が極大となる位置を求める手法である．mean shift 法による追跡では，前フレームでの検出位置の近傍で，追跡対象と最も類似した位置を求めることができる [14]．これを動画像の各フレームに対して適用することにより，追跡を行う．

mean shift 法では，探索窓内の密度の偏りがあると，その偏りが移動オブジェクトによるものであろうと妨害オブジェクトによるものであろうと，その偏りを追跡する．そのため，移動オブジェクトが 1 度探索窓に入ってしまうと，その偏りを追跡してしまうので，結果として移動オブジェクトを追跡することとなる．

提案方式の移動オブジェクトを mean shift 法を用いて追跡する様子を**図 11** に示す．mean shift 法では，探索窓内の確率分布の重心を計算して，その位置に探索窓の中心が来るように探索窓の位置を更新するという手順で物体の追跡を行う．提案する CAPTCHA の移動オブジェクトの初期位置はランダムのため，探索窓の初期位置は画面の中央に置く．提案方式において，妨害オブジェクトはランダムに出現して消え，移動オブジェクトは図 8 に示したとおり，連続的に移動する．

そこで本提案手法では，追跡精度を低く抑えるために，妨害オブジェクトの数を増やすこととした．移動オブジェクトを捕捉した探索窓内に妨害オブジェクトが入り込むと，それ以降は誤って妨害オブジェクトを追跡してしまうことがある．妨害オブジェクトの増加により，この確率が高まることを期待したのである．しかし，妨害オブジェクトを増やすと，人間にとって見づらいものになる危険性がある．そこで提案する CAPTCHA を現実のものとするには，妨害オブジェクト数を増加させたときのボットの追跡精度と人間の追跡精度がどのように変化するかを確認しなければならない．よって，その性能評価を実証的な実験により行う（5 章）．

4. リレーアタック耐性の検証実験

提案手法の CAPTCHA に対してリレーアタックを行い，リレーアタックで生じる遅延時間によって，CAPTCHA の解答が困難になることを確認する．このとき，幇助ユーザと中継 PC 間の通信の遅れ時間は両者間の通信環境に依存して様々であるため，複数の異なる遅延時間を発生させて実験を行う．

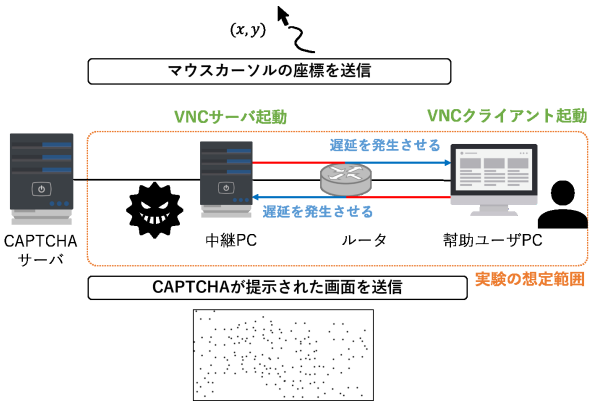

図 12 実験環境
Fig. 12 Experimental environment.

さらに，正規のアクセスとリレーアタックによるアクセスそれぞれの追跡成功時間の結果を用いて，この 2 つを判別する閾値について検討する．この閾値を大きく設定すれば，リレーアタックを受け入れてしまう誤判定を減らせるが，正規のアクセスを拒絶してしまう誤判定が増えてしまう．すわなち，正規ユーザ拒否率（FRR：False Rejection Rate）が大きくなってしまう．逆に小さく設定すれば，正規のアクセスを拒絶してしまう誤判定は減らせるが，リレーアタックを受け入れてしまう誤判定が増えてしまう．すなわち，幇助ユーザ受容率（FAR：False Acceptance Rate）が大きくなってしまう．

具体的には，まず，実験から得られた双方の追跡成功時間のデータから，それぞれのアクセスの追跡成功時間の確率分布を推定する．この確率分布を利用することにより，閾値の値を様々に変えたときの正規ユーザ拒否率と幇助ユーザ受容率を試算できるようにしておく．そのうえで，正規アクセスの成功率が代表的な CAPTCHA の 1 つである google の reCAPTCHA ver.1 と同程度になるように閾値を選ぶ．最後に，その閾値の下で推算される幇助ユーザ受容率では，幇助ユーザが期待できる成功報酬が低く，リレーアタックが成立するとは考えられないことを示す．

4.1 実験方法

本実験の実験環境の概略を**図 12** に示す．CAPTCHA のサーバと中継 PC を同一 PC においている理由は，本実験の目的が中継 PC と幇助ユーザ PC の間の通信遅延時間がもたらすリレーアタックへの影響を調べることであり，CAPTCHA サーバと中継 PC の遅延はそれに比べて小さいと想定していることから，このような簡易な実装で問題がなかったからである．

実験では，10 名の実験参加者（21 歳～25 歳）に，正規

のアクセスを模した中継 PC 上での CAPTCHA の解答と，リレーアタックを模した幇助ユーザ PC での CAPTCHA の解答を，それぞれ 5 回ずつ実施させて移動オブジェクトの追跡成功時間を計測した．

4.2 実験環境

リレーアタックの再現は，VirtualBox を利用し仮想環境上で行った．VirtualBox は，既存のオペレーティング・システム（ホスト OS）上にアプリケーションの 1 つとしてインストールされ，この中で追加のオペレーティング・システム（ゲスト OS）を実行することができる．実験では，中継 PC と幇助ユーザ用の PC をゲスト OS として用意し，2 つのゲスト OS 間で CAPTCHA の中継を行った．CAPTCHA の中継には文献 [7] でリレーアタックを再現するためのソフトウェアとして利用されていた VNC（Virtual Network Computing）を用いた．VNC は，ネットワークを通じて接続された他のコンピュータの画面を遠隔操作できるソフトウェアである．遅延時間の発生には，VyOS [15] を用いた．VyOS は，オープンソースで開発されているネットワーク OS であり，主にソフトウェアルータとして運用される．中継 PC と幇助ユーザ PC 間の Round Trip Time（通信相手にデータを送信して応答が帰ってくるまでにかかる時間）を約 50 ms，100 ms，200 ms になるように設定した．たとえば東京にサーバがある場合，50 ms の遅延が発生するのは，日本国内や韓国や台湾などのアジアの中でも東京に近い地域である．日米間で考えると，西海岸までは 100 ms 程度，東海岸までは 200 ms 程度とされている [16]．計 3 パターンの通信環境でのリレーアタックを行い，追跡成功時間のデータを収集した．実験環境の諸元は，以下のとおりである．

中継 PC，幇助ユーザ PC（ゲスト OS）：Ubuntu 16.04 LTS

VNC サーバー：Vino

VNC クライアント：Remmina（色数は「256 色」，品質は「最高」に設定）

CAPTCHA のパラメータ：妨害オブジェクトの数（20 個），移動オブジェクトの速さ（毎フレーム，0.2 ピクセル～7.0 ピクセルの移動量），フレームレート（60 fps）

色数の 256 色は最低の設定値である．これは，本提案方式に色数は影響しないため，通信の負荷を抑える設定にしたためである．一方，品質については最高の品質に設定した．これは，本提案方式がリアルタイムにオブジェクトを追跡する手法となっているために画面の更新が十分に高速に行われる必要があるからである．

4.3 実験結果と考察

実験の結果から，通信の遅延により，リレーアタックの効果を減じられることが確認できた．本実験で，観測された正規アクセスでの追跡成功時間と 4 パターンの通信環下でのリレーアタックの追跡成功時間の結果を**図 13**，**図 14**，**図 15**，**図 16** に示す．通信の遅延が大きくなるにつれ，リレーアタックの追跡成功時間が短くなる傾向がある．

次に，得られた追跡成功時間のデータを使用し，リレー

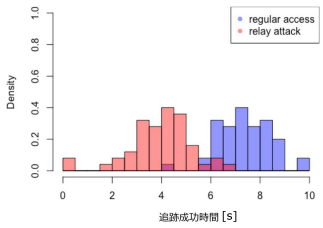

図 13 正規アクセスとリレーアタック（遅延の設定なし）の追跡成功時間

Fig. 13 No insertion of delay time.

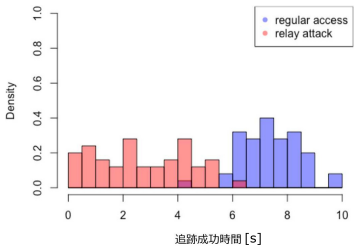

図 14 正規アクセスとリレーアタック（50 ms の遅延）の追跡成功時間

Fig. 14 Insert delay time of 50 ms.

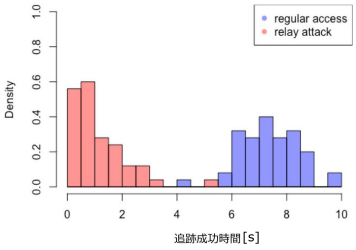

図 15 正規アクセスとリレーアタック（100 ms の遅延）の追跡成功時間

Fig. 15 Insert delay time of 100 ms.

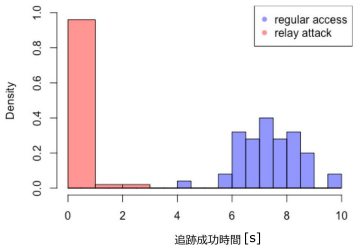

図 16 正規アクセスとリレーアタック（200 ms の遅延）の追跡成功時間

Fig. 16 Insert delay time of 200 ms.

表 1 追跡成功時間の正規性の検定の結果
Table 1 The result of the normality test of data of the tracking success time.

	p value
正規アクセス	0.2728
遅延設定なし	0.0667
遅延時間 50 ms	0.0626

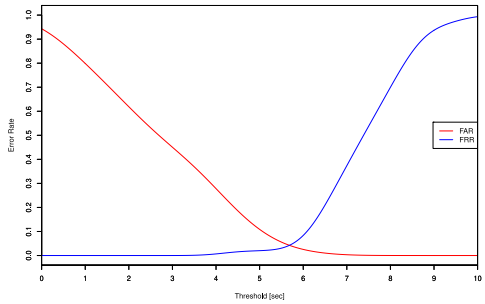

図 17 遅延 50 ms：幇助ユーザ受容率（FAR）：赤線と正規ユーザ拒否率（FRR）：青線
Fig. 17 FAR and FRR (Delay 50 ms).

アタックの達成が困難となる様，追跡成功時間の閾値を選択することを試みる．提案手法の考え方の下では，遅延の大きさがより小さい条件で得られた閾値は，遅延がより大きい条件でも有効であると考えられる．そこで，今回の実験で採用した 3 種類の遅延時間のうち，最小の 50 ms の場合の実験結果を対象に閾値の探索を行うこととした．

最初に，観測されたデータを元に，それぞれのアクセスの成功率の確率分布を推定する．今回は正規分布に従うものと仮定することとし，まず，実験結果が正規分布で近似できるかどうか，シャピロウィルク検定を行った．シャピロウィルク検定 [17] では，標本分布は，正規分布に従うという帰無仮説を検定する．検定結果の指標は，p 値（有意確率）を用い，有意水準 5%で検定を行うため，$p > 0.05$ となれば，正規分布に従うと判断できる．

検定の結果，正規アクセスの追跡成功時間と遅延時間 50 ms のリレーアタックの追跡成功時間は，正規分布に従うことが分かった．検定結果は**表 1** に示すとおりである．

次に，正規アクセスと遅延時間 50 ms のリレーアタックの追跡成功時間のデータを近似した正規分布から，幇助ユーザ受容率（FAR：False Acceptance Rate）と正規ユーザ拒否率（FRR：False Rejection Rate）を算出した．得られた FAR と FRR を**図 17** に示す．

この図を利用して，CAPTCHA として実用的な閾値を考える．文献 [18] によれば，一般的に利用されている google の reCAPTCHA ver.1 の平均成功率は，97%である．今回の提案手法の CAPTCHA において，正規ユーザの成功率をこれと同程度にしたい場合は，FRR が 3%になるように閾値を設定すれば良い．このときの追跡成功時間の閾値は約 5.5 秒である．

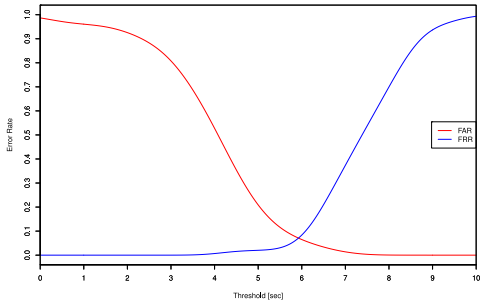

図 18 遅延設定なし：幇助ユーザ受容率（FAR）：赤線と正規ユーザ拒否率（FRR）：青線
Fig. 18 FAR and FRR (No insertion of delay).

一方，閾値を上述の値とした場合，FAR は図 17 から約 5%と読み取れる．幇助ユーザは，CAPTCHA 1,000 個の解読につき報酬を得ているが，FAR が 5%であれば，1,000 個解いたとしても，このうち成功するのは，50 個程度となるため，幇助ユーザにとっては，かけた時間に対して得られる報酬額が十分でなく，提案手法に基づく CAPTCHA を解く仕事は請け負わないものと考えられる．この状況にもかかわらず，十分な数の幇助ユーザを確保するには，単価をあげる必要があるが，雇用する側にとっては受け入れられないものと考えられる．また，単価をあげてまでアクセスしたい特別な価値のある Web サイトの数は多いとは考えられず，攻撃者自らが解けば十分であるので，そもそも依頼が行われないと考えられる．

また，100 ms，200 ms と遅延が大きくなるにつれて，幇助ユーザの追跡成功時間が短くなっていることから，ここで設定した閾値は，50 ms より大きい遅延が発生するリレーアタックに対しても有効であると考えられる．また，正規ユーザがどこからアクセスしているかが分かれば，そこから逆算して最適な閾値を設定することができると考えられる．ただし遅延が 50 ms より小さいときには，十分な効果を発揮できるとはいえない．**図 18** に，意図的な遅延を発生させない（遅延時間 0 ms）場合の，FAR と FRR を示す．ネットワーク遅延がない場合は，この図より，FRR が 3%になるように設定した場合の追跡成功時間の閾値は，50 ms の場合とほぼ同じで約 5.5 秒であるが，このときの FAR は 10%となってしまい，遅延 50 ms の場合に比べ，幇助ユーザを受け入れてしまう確率が高くなってしまう．

本研究では CAPTCHA サーバと正規ユーザ間は，低遅延であることを想定している．たとえば，CAPTCHA サーバと正規ユーザや中継 PC は日本国内にあり，中継 PC と外国にいる幇助ユーザ PC の間に 50 ms 以上の遅延がある場合に提案方式は有効であるといえる．この想定に反して，CAPTCHA サーバと正規のユーザの間に遅延が生じた場合は正規のユーザであっても提案 CAPTCHA の成立が難しくなる．

今後の課題として，リレーアタックの評価環境として，

CAPTCHA サーバと中継 PC を同一 PC 上におかず，それぞれの模擬システムをネットワークで結合した異なる PC 上に置くようにし，CAPTCHA サーバと中継 PC 間に遅延が生じた場合の提案手法の性能を評価する必要がある．

5. ボット耐性と追跡精度の検討

本章では，まず，目標とするボットの受容率が与えられたときに，それを達成するために必要な追跡成功時間が何秒であるかを，人間とボットに CAPTCHA を解かせた結果から求められるようにする．さらに，適切な閾値を選ぶことができれば，提案する CAPTCHA が既存の代表的な CAPTCHA との比較のうえで十分なボット耐性を持つことを示す．

また，妨害オブジェクト数の増加が，ボットの追跡精度と人間の追跡精度にどの程度の影響をもたらすのかを調べ，具体的な値を示す．

5.1 実験のためのデータ収集

20 名の被験者（21 歳～24 歳）に提案手法の CAPTCHA を解かせ，移動オブジェクトの追跡成功時間を測定する．CAPTCHA が表示する妨害オブジェクトの数については，10 個，20 個，30 個，40 個，50 個の 5 パターンを用意し，それぞれを 10 回ずつ解かせる．被験者が 20 名であるので，人間の追跡成功時間のデータが，各パターンにつき 200 個ずつ集められる．

この実験では，mean shift 法を用いた物体追跡のプログラムを実装し，マウスカーソルが自動的に移動オブジェクトを追跡するボットを作成した．このプログラムの実装には，画像処理ライブラリの OpenCV [19] を用いた．

5.2 ボットに対する安全性の評価・考察

まず，計測されたデータを元に人間とボットそれぞれの追跡成功時間の確率分布を推定し，ボット受容率（FAR）と人間拒否率（FRR）を与えたときにそれらを満たす閾値を算出できるようにした．ここでは，カーネル密度推定を用いて，確率分布を算出した．その結果を，**図 19**，**図 20**，**図 21**，**図 22**，**図 23** に示す．これらの図には，収集したデータのヒストグラムと，算出した確率密度曲線を描画している．

ここから提案 CAPTCHA のボット耐性について考察するにあたり，上述の 5 種類の妨害オブジェクト数のうち，人間とボットの差が顕著であった，50 個の場合のデータに基づいて議論することにする．**図 24** に，妨害オブジェクト 50 個の場合の FAR（ボット受容率）と FRR（人間拒否率）を示す．

まず，CAPTCHA のセキュリティ目標を 1%以下のボット受容率（CAPTCHA 1,000 個の場合のボットによる解読数 10 個）とした場合について検討する．この目標値は，

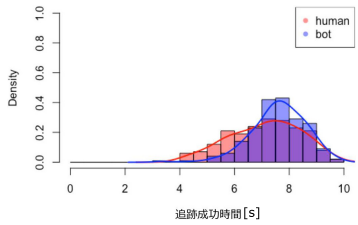

図 19 人間とボットの追跡成功時間（妨害オブジェクト 10 個）
Fig. 19 The num. of decoy objects is 10.

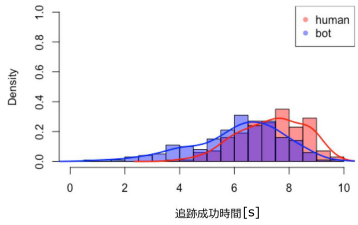

図 20 人間とボットの追跡成功時間（妨害オブジェクト 20 個）
Fig. 20 The num. of decoy objects is 20.

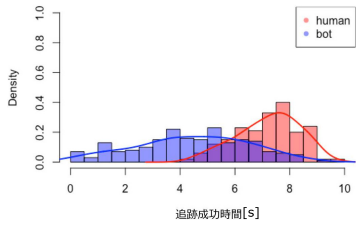

図 21 人間とボットの追跡成功時間（妨害オブジェクト 30 個）
Fig. 21 The num. of decoy objects is 30.

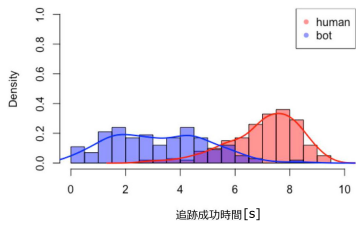

図 22 人間とボットの追跡成功時間（妨害オブジェクト 40 個）
Fig. 22 The num. of decoy objects is 40.

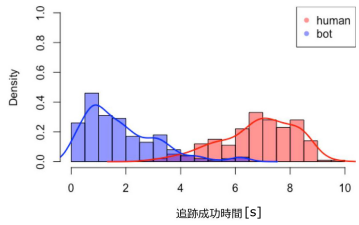

図 23 人間とボットの追跡成功時間（妨害オブジェクト 50 個）
Fig. 23 The num. of decoy objects is 50.

[20], [21] において，ボットの突破率は 1%を超えてはいけないとされていることから選んだ値である．このときの追跡成功の閾値となる追跡時間は，図 24 から約 6.2 秒である．この場合，FRR（人間拒否率）は約 30%になり，妨害オブジェクト数の増加により人間の追跡精度が低下してしまっていることが懸念される．

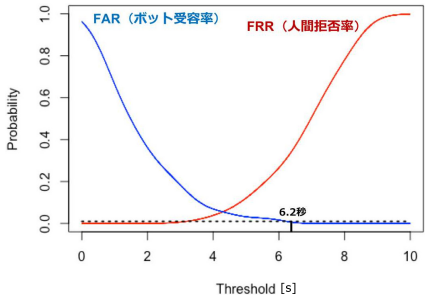

図 24 FAR（ボット受容率）：青線と FRR（人間拒否率）：赤線（妨害オブジェクト 50 個）
Fig. 24 FAR and FRR (The num. of decoy objects is 50).

表 2 既存手法との比較
Table 2 The comparison with existing methods.

分類	文字列	画像	動画
	reCAPTCHA v1	reCAPTCHA v2	提案手法
FRR	0.03 [18]	0.159 [22]	0.099
FAR	0.874 [23]	0.707 [24]	0.036
平均回答時間	8.55 秒 [22]	6 秒未満 [22]	10 秒程度

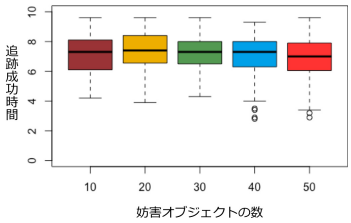

図 25 人間の追跡成功時間
Fig. 25 The tracking success time (human).

一方，FRR（人間拒否率）を 10%未満にすると追跡成功時間の閾値は，図 24 から約 4.8 秒である．このときの FAR（ボット受容率）は 3.6%であり，文献 [20], [21] の目標値は達成できていないが，既存の代表的な CAPTCHA と比較すると（**表 2**）十分なボット耐性を有している．なお，マウスでのクリック操作を必要とする google reCAPTCHA ver.3 についても一部報告 [25] があり，比較の対象に含めることも検討したが，仕様に不明な点が多いため除外した．

4 章では妨害オブジェクト数が 20 個でもリレーアタック耐性があったことを考えると，妨害オブジェクト数を 50 個に増やしても，耐性が増すことはあっても損なわれることがあるとは考えにくい．このことから，妨害オブジェクト数が 50 個あれば，リレーアタックとボット耐性の両立できるといえる．

5.3 ボットと人間の追跡精度に関する検討

次に妨害オブジェクトの数を増やした場合でも人間の追跡成功時間にあまり影響がないことを確認する．人間の追跡成功時間のデータを**図 25**，ボットの追跡成功時間のデータを**図 26** に示す．図では，妨害オブジェクトの数を変化させたときの追跡成功時間の箱ひげ図を示している．

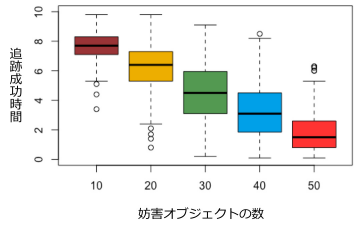

図 26 ボットの追跡成功時間
Fig. 26 The tracking success time (bot).

第 1 四分位点（中央値より小さい値に限定したデータの中央値）から第 3 四分位点（中央値より大きい値に限定したデータの中央値）までの高さに箱を描き，箱から上下に伸びている黒線（ひげ）は，それぞれ最大値，最小値を示している．白丸で描画しているのは，外れ値と見なされたデータである．

図 25 と図 26 を見ると，妨害オブジェクトの増加にともない，ボットの追跡成功時間は低下していき，人間の追跡成功時間は，変化がほとんどない．このことから，妨害オブジェクトの増加によるボットへの影響は大きく，人間への影響は少ないことが分かった．

平均回答時間について評価すると，google reCAPTCHA v1 では平均 8.55 秒，google reCAPTCHA v2 では 6 秒未満と報告されているのに対し，本提案方式は，必ず 10 秒の追跡を行う方式であるため，平均回答時間は 10 秒程度となってしまう（表 2）．ただし，ユーザの追跡成功時間が閾値を超えた時点で終了させるという改良が考えられる．

これらの結果より，CAPTCHA のセキュリティ目標を達成しつつ，ユーザビリティを低下させないためには，妨害オブジェクトを増やすことが有効であることが分かった．ただし，移動オブジェクトを見つけるのに，時間がかかる人もおり，まったく人間に影響がないわけでないため，セキュリティを保ちつつ，人間のユーザビリティを低下させない適切なパラメータの決定が必要だと考えられる．

6. Deep Learning 技術の適用実験

本章では，動画を対象とした Deep Learning 技術を適用することにより，リアルタイムでの移動オブジェクト追跡が可能であるかどうかを確認する実験を行った．具体的には，1 秒あたりのフレーム数と妨害オブジェクト数を様々に変えて実験し，リアルタイムでの追跡が可能なフレーム数やそのときのオブジェクト数に応じた追跡成功時間についての検討を行う．

6.1 使用する Deep Learning 技術

具体的な Deep Learning 技術としては，Faster R-CNN [26] を採用した．提案する CAPTCHA では，移動オブジェクトや妨害オブジェクトが互いに重なりあうことが多く，それらを分離して認識する必要がある．Faster

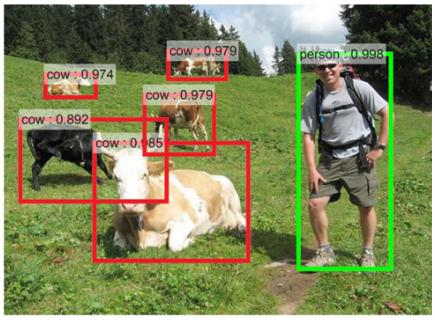

図 27 Faster R-CNN の認識結果*5
Fig. 27 Recognition result of Faster R-CNN.

R-CNN は，**図 27** に示すように，たとえ対象物が重なっていたとしても物体が存在する場所を取得することができ，これによって対象物を認識することができることが，その採用理由である．Faster R-CNN は物体存在箇所を特定するネットワークと物体識別の 2 つのネットワークで構成される Deep Learning 技術である．これら 2 つのネットワークは，それぞれ Region Proposal Network（RPN）と Convolutional Neural Network（CNN）により構成される．

6.2 Deep Learning 技術を適用した追跡手法

本実験では，オブジェクトの認識に Deep Learning 技術を用いる．オブジェクトが認識できた後は，前述の mean shift 法を用いた追跡手法と同様，フレーム間での位置の変化を利用して移動オブジェクトを特定するアプローチを用いる．すなわち，妨害オブジェクトはフレームごとにランダムに位置を変えるが，移動オブジェクトは連続的に移動する特徴を利用する．

Deep Learning 技術を用いた物体認識は次のように行う．まず，認識したい物体の学習を行う．認識したい物体それぞれ表す解答データを用意し，また，物体の存在箇所を特定するために，いくつかの形の異なる Anchor Box（図 27 の赤や緑の枠に該当）を用意する．そして適当に Anchor Box をあてはめ，ある閾値以上に解答データに重なる Box を正例，どの学習データにも重ならない Box を負例として学習を行う．本実験では，図 8 に示す，円弧状に並んだ点群（今回の場合 2 パターン）を解答データとした（移動オブジェクトと妨害オブジェクトで，同一のデータである）．この学習により，CAPTCHA 動画の各フレームにおいて，移動オブジェクトと妨害オブジェクトの認識が可能となる．

次に，学習した識別器を用いて，以下の手順で追跡を行う．

(1) Deep Learning 技術を用いて画像を解析し，オブジェクト領域を抽出する．
(2) 2 時点前，3 時点前の画像も同じく解析しオブジェクト領域を抽出する．
(3) 3 画像のオブジェクト領域を AND 演算して，重なる領域を抽出する．
(4) 重なった領域のうち最大領域を求める（輪郭抽出処理）．
(5) 最大領域の重心を追跡対象オブジェクトの座標とする．
(6) 存在しなかった場合は直前の位置と同じ位置とする．
(7) 出力した追跡オブジェクトの座標が，正解の範囲（3.3 節の式 (1)）に入っていれば追跡成功とする． □

*5 文献 [26] の図 5 より転載

表 3 処理時間（秒）
Table 3 Processing time (sec).

	妨害オブジェクト数				
FPS	10	20	30	40	50
15	84.487	87.504	87.029	87.542	87.475
30	170.592	171.102	179.915	171.015	171.363
45	253.823	254.114	254.745	254.566	255.065
60	337.203	338.171	337.259	337.536	337.563

6.3 実験条件

実験用のプログラムは文献 [27] をベースに開発を行った．学習が高速に行えるように，GPU を用いるよう設定し，Faster R-CNN の物体識別用のニューラルネットワークには ResNet を適用した．実験のパラメータとして，Anchor Box の種類を複数設定できるが，本実験で認識したい 2 種類のオブジェクト（移動オブジェクトと妨害オブジェクト）は互いに同一の大きさで同一の形状であるため，単一の Anchor Box で学習を行わせた．また，学習を効率良く行えるようにするために，パラメータの初期値として，あらかじめ学習済みモデルの値が利用できるようになっているが，本実験で対象とするような，2 種類の同一形状のオブジェクトを対象としたモデルはなかったので，バッチ学習により一から学習を行った．

本実験では，10,000 枚のフレーム画像を用意して学習を行わせた．その後に，FPS を 15，30，45，60 の 4 通り，妨害オブジェクト数を 10，20，30，40，50 の 5 通りに変えて，20 の条件で実験を行った．

6.4 実験結果

FPS と妨害オブジェクト数を様々に変えて実験した結果として，オブジェクトの認識に要した処理時間を**表 3** に示す．FPS を最小にした条件下であっても，10 秒間の画像を処理するのに 10 秒を超える時間が必要であり，ここで採用した Deep Learning 技術による物体認識手法では，リアルタイムでの物体追跡が行えなかった．また，表 3 に示す処理時間の値のうちの大部分が，Deep Learning 技術を用いてオブジェクト領域を抽出する処理に要した時間であった．なお，処理に要する時間は，FPS が大きくなると，それにほぼ比例して大きくなり，妨害オブジェクト数には依存しなかった．

次に，移動オブジェクトを認識して追跡に成功していた

表 4 追跡成功時間（秒）
Table 4 Tracking Success time (sec).

	妨害オブジェクト数				
FPS	10	20	30	40	50
15	8.349	6.333	4.447	2.593	1.947
30	7.230	6.260	4.660	3.033	1.970
45	8.218	6.240	4.744	3.498	2.044
60	8.270	6.458	4.583	3.105	1.940

時間の長さを**表 4** に示す．ここでの時間の長さとは，10秒分のフレーム中で，移動オブジェクトの正しい位置を得ることができていたフレームの割合の意味であり，たとえば 90%のフレームで成功していれば，追跡成功時間を 9.0 秒とした．mean shift 法と比べると，追跡成功時間はやや向上した．またどの条件下でも，妨害オブジェクト数が大きくなるにつれて追跡成功時間は短くなり（おおよそ反比例），FPS には依存しなかった．

3.5.3 項で考察したとおり，今回の提案方式では CAPTCHA に出現する 2 種類のオブジェクト，移動オブジェクトと妨害オブジェクトは同一形状で同一の大きさであり，一般的な Deep Learning 技術の適用の仕方（ある種のオブジェクトを，それとは異なるオブジェクト群から識別する）では効果が発揮できないと考えられる．

7. まとめ

本論文では，CAPTCHA に対する攻撃の代表例であるリレーアタックとボットによる CAPTCHA の脆弱性を突いた自動的な攻撃への耐性を両立するためのインタラクティブな動画 CAPTCHA 方式を提案した．本手法は，連続的に移動するオブジェクトをリアルタイムにマウスカーソルで追跡し，その追跡が成功していた時間で人間かボットかを判別する．

評価として，リレーアタックをシミュレートするための実験環境を構築し，提案手法に対してリレーアタックを行い，ある一定の耐性を持つことを確認した．さらに，ボット耐性については mean shift 法と Deep Learning 技術を実装して評価した．前者の評価から，難読化の度合いを強めることで実装した攻撃への耐性は強化されることが分かった．また，これにともなう人間への影響はあまり見られなかった．後者の評価から，一般的な Deep Learning 技術の適用の仕方では効果が発揮できないことが分かった．

今後の課題として，VPN の使用や Tor 経由の場合など，様々なネットワーク環境下でのリレーアタックの検証や，幅広い年齢層のユーザビリティ評価が必要だと考えられる．さらに，ボット対策の改良やそれによりユーザビリティにどのような影響があるかをあわせて調査する必要がある．

謝辞 本研究は JSPS 科研費 JP18K11268，JP21K11849 の助成を受けたものです．

参考文献

[1] Von Ahn, L., Blum, M., Hopper, N. and Langford, J.: Telling humans and computers apart automatically, *Comm. ACM*, Vol.47, pp.50–60 (2004).
[2] Inaccessibility of CAPTCHA, available from ⟨https://www.w3.org/TR/turingtest/⟩ (accessed 2020-09-16).
[3] Yan, J. and El Ahmad, A.S.: Breaking visual CAPTCHAs with naive pattern recognition algorithms, *the 23rd Annual Computer Security Applications*, pp.279–291, IEEE Computer Society (2007).
[4] Chellapilla, K. and Simard, P.Y.: Using machine learning to break visual human interaction proofs (HIPs), *Advances in Neural Infomation Processing Systems*, Vol.17, pp.265–272 (2005).
[5] Elson, J., Douceur, J.D., Howell, J. and Saul, J.: Asirra: A CAPTCHA that exploits interest-aligned manual image categorization, *Proc. 14th ACM Conference on Computer and Communications Security* (*CCS*) (2007).
[6] Mohamed, M., Sachdeva, N., Georgescu, M., Gao, S., Saxena, N., Zhang, C. and Chen, W.B.: A three-way investgation of a game-CAPTCHA: Automated attack, relay attacks and usability, *Proc. 9th ACM Symposium on Information, Computer and Communications Srcurity*, pp.195–206, ACM (2014).
[7] Mohamed, M., Gao, S., Saxena, N. and Zhang, C.: Dynamic cognitive game captcha usability and detection of streaming-based farming, *the Workshop on Usable Security* (*USEC*), co-located with NDSS (2014).
[8] Gao, S., Mohamed, M., Saxena, N. and Zhange, C.: Gaming the game: Defeating a game captcha with efficient and robust hybrid attacks, *2014 IEEE International Conference Multimedia and Expo* (*ICME*), pp.1–6, IEEE (2014).
[9] 3D-based Captchas become reality, clnet (online), available from ⟨http://ascii.jp/elem/000/000/483/483759/index-2.html⟩ (accessed 2020-09-16).
[10] Tatsuda, R., Aburada, K., Yamaba, H., Katayama, T., Mukunoki, M., Park, M. and Okazaki, N.: An examination of the interactive video CAPTCHA method to resist relay attack, *IEICE Communications Express*, Vol.7, No.4, pp.136–141 (2018).
[11] Aburada, K., Usuzaki, S., Yamaba, H., Katayama, T., Mukunoki, M., Park, M. and Okazaki, N.: An evaluation of the interactive video CAPTCHA method against automated attack, *IEICE Communications Express*, Vol.8, No.12, pp.453–457 (2019).
[12] Khan, I.R. and Farbiz, F.: A back projection scheme for accurate mean shift based tracking, *2010 17th IEEE Int. Conf. Image Processing* (*ICIP*), pp.33–36 (2010).
[13] Cheng, Y.: Mean shift, mode seeking, and clustering, *IEEE Trans. PAMI*, Vol.17, No.8, pp.790–799 (1995).
[14] Comaniciu, D., Ramesh, V. and Meer, P.: Real-time tracking of non-rigid objects using mean shift, *Proc. IEEE Conf. CVPR*, pp.142–149 (2000).
[15] Index of /software/vyos/iso/release/1.1.7 (online), available from ⟨http://ftp.tsukuba.wide.ad.jp/software/vyos/iso/release/1.1.7/⟩ (accessed 2020-09-16).
[16] Fontugne, R., Mazel, J. and Fukuda, K.: An Empirical Mixture Model for Large-Scale RTT Measurements, *2015 IEEE Conference on Computer Communications* (*INFOCOM*), pp.2470–2478 (2015).
[17] Shapiro, S.S. and Wilk, M.B.: An analysis of variance test for normality (complete samples), *Biometrika*,

Vol.52, No.3, pp.591–611 (1965).
[18] Yan, J. and El Ahmad, A.S.: Usability of CAPTCHAs or usability isuues in CAPTCHA design, *Proc. 4th Sympousium on Usable Privacy and Security*, pp.44–52, ACM (2008).
[19] Meanshift and camshift, available from ⟨https://docs.opencv.org/3.4/d7/d00/tutorial_meanshift.html⟩ (accessed 2020-09-16).
[20] El Ahmad, A.S., Yan, J. and Ng, W.Y.: CAPTCHA design: Color, usability, and security, *IEEE Internet Computing*, Vol.16, No.2, pp.44–51 (2012).
[21] Chellapilla, K., Larson, K., Simard, P.Y. and Czerwinski, M.: Building segmentation based human-friendly Human Interaction Proofs (HIPs), *Human Interactive Proof*, Vol.3517, pp.1–26 (2005).
[22] Jiang, N., Dogan, H. and Tian, F.: Designing mobile friendly CAPTCHAs: An exploratory study, *31st British Human Computer Interaction Conference 2017*, Vol.92, pp.1–7 (2017).
[23] Guixin, Y., Tang, Z., Fang, D., Zhu, Z., Feng, Y., Xu, P., Chen, X. and Wang, Z.: Yet another text captcha solver: A generative adversarial network based approach, *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp.332–348 (2018).
[24] Sivakorn, S., Polakis, I. and Keromytis, D.A.: I am robot: (deep) learning to break semantic image CAPTCHAs, *IEEE European Symposium on Security and Privacy*, pp.388–403 (2016).
[25] Akrout, I., Feriani, A. and Akrout, M.: Hacking google reCAPTCHA v3 using reinforcement learning, arXiv:1903.01003v3 (2019).
[26] Ren, S., He, K., Girshick, R. and Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.39, No.6, pp.1137–1149 (2017).
[27] keras_frcnn, available from ⟨https://github.com/jinfagang/keras_frcnn⟩ (accessed 2021-06-30).

油田 健太郎 （正会員）

2003 年宮崎大学工学部情報システム工学科卒業．2005 年同大学大学院工学研究科情報工学専攻博士前期課程修了．2006 年熊本県立大学総合管理学部助手．2009 年宮崎大学大学院工学研究科システム工学専攻博士後期課程修了．同年大分工業高等専門学校助教．2012 年同講師．2017 年宮崎大学工学部情報システム工学科准教授．博士（工学）．コンピュータネットワークに関する研究に従事．電子情報通信学会，電気学会各会員．

臼崎 翔太郎 （学生会員）

2017 年宮崎大学工学部情報システム工学科卒業．2017 年同大学大学院工学研究科工学専攻博士前期課程修了．2020 年同大学院農学工学総合研究科物質・情報工学専攻博士後期課程在籍．ネットワークセキュリティに関する研究に従事．

山場 久昭 （正会員）

1988 年東京工業大学工学部化学工学科卒業．1990 年東京工業大学大学院総合理工学研究科化学環境工学専攻修士課程修了．同年花王（株）入社．1993 年宮崎大学工学部助手．2007 年同助教．2010 年宮崎大学大学院博士後期課程システム工学専攻単位取得満期退学．博士（工学）．生産システム設計・運用の計算機支援に関する研究に従事．化学工学会，人工知能学会，計測自動制御学会各会員．

片山 徹郎 （正会員）

1991 年九州大学工学部情報工学科卒業．1993 年同大学大学院工学研究科情報工学専攻修士課程修了．1995 年同大学院工学研究科情報工学専攻博士後期課程修了．同年奈良先端科学技術大学院大学情報科学研究科助手．2000 年宮崎大学工学部情報システム工学科助教授．2007 年同大学准教授を経て 2018 年同大学教授．博士（工学）．ソフトウェア工学，特にソフトウェアのテスト技法や信頼性に関する研究に従事．電子情報通信学会，日本ソフトウェア科学会各会員．

椋木 雅之 （正会員）

1991 年京都大学工学部情報工学科卒業．1996 年同大大学院工学研究科博士後期研究指導認定退学．同年同大学工学部助手．1998 年同大学院情報学研究科助手．2000 年同大学総合情報メディアセンター助手．2002 年同大学学術情報メディアセンター助手．同年広島市立大学情報科学部助教授．2007 年同大大学院情報科学研究科准教授．2009 年京都大学学術情報メディアセンター准教授，2015 年宮崎大学工学部情報システム工学科教授．博士（工学）．画像認識，コンピュータビジョン，映像メディア処理の研究に従事．電子情報通信学会，画像電子学会，IEEE 各会員．

朴 美娘 （正会員）

1983 年漢陽大学工学部電子工学科卒業．同年同大学工学部助手．1993 年東北大学大学院工学研究科情報工学専攻博士後期課程修了．同年同大学電気通信研究所助手．1994 年三菱電機株式会社入社．2010 年神奈川工科大学情報学部教授．博士（工学）．ネットワークセキュリティ，暗号プロトコル設計，認証等の研究に従事．IEEE，電子情報通信学会，日本セキュリティ・マネジメント学会各会員．

岡崎 直宣 （正会員）

1986 年東北大学工学部通信工学科卒業．1991 年同大学大学院工学研究科電気および通信工学専攻博士後期課程修了．同年三菱電機株式会社入社．2002 年宮崎大学工学部助教授．2007 年同大学准教授を経て 2011 年同大学工学教育研究部教授．博士（工学）．通信プロトコル設計，ネットワーク管理，ネットワークセキュリティ，モバイルネットワーク等の研究に従事．電子情報通信学会，IEEE 各会員．