Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

人間の持つ高度な知覚的補完能力を利用した画像型 CAPTCHA の提案

上松晴信† 松井利樹† 三宅優†

近年，インターネット上ではボットによるフリーメールアカウントの不正取得やブログに対するスパムコメントの被害が増加している．このような被害を抑えるために人間が持つ高度な画像認識特性を利用した画像型 CAPTCHA が幾つか提案されているが，問題の自動生成などの実用的な観点から課題が残っている．本論文では問題の自動生成が可能で，かつ人間にも解くことが出来る CAPTCHA を提案する．人間は断片的な情報を知覚的に補完し意味のある情報へと復元することが出来るが，これは機械にとっては難しい．この人間特有の知覚的補完能力を利用して，複数の円画像間の関係性を問う CAPTCHA を構築した．機械が推測可能か確かめるために画像解析プログラムを作成して評価を行い，人間が解いた際の正答率と解答時間を計測して考察を行った．

The proposal of image based CAPTCHA using the human perception complement ability

HARUNOBU AGEMATSU† TOSHIKI MATSUI†
YUTAKA MIYAKE†

Recently, the number of immoral acquisition of free mail accounts and spam comments against blog by bot are increasing. To prevent such situations, some image based CAPTCHAs using human image recognition ability are proposed. But there is a problem in practical use like automatic generation of question. We propose a new CATPCHA that can generate questions automatically and human can solve easily. Human can restore the original information from fragmented information but this is too difficult for bot programs. We developed the CAPTCHA that asked human the relationship of the several circle images, and the human solved the problem using this human perception complement ability. We made the image analyzing program to ensure whether bot programs can solve or not and discuss about the percentage of correct answers and response time.

1. はじめに

近年，インターネット上では自動プログラムによって，フリーメールのアカウントを大量に不正取得する，ブログサイトにスパムコメントを不正投稿する等の DoS（Denial of Service，サービス不能）攻撃が頻発している．チューリングテストは，そのサービスのユーザが人間であるか機械（ボット）であるかを判別するための技術であり，CMU の研究者によって開発された CAPTCHA(Completely Automated Public Turing Test To Tell Computers and Humans Apart)[1]と呼ばれる技術が，このような DoS 攻撃を防ぐために広く普及している．

現在の主流は文字型 CAPTCHA と呼ばれているもので，文字を歪ませて人間に識別させるものである．これは人間の高い文字認識能力を利用したものであったが，すでに高機能な OCR（自動文字読取）機能を備えるボットが出回るようになっている[2]. 文字列に加える変形やノイズを大きくすることによってボットを排除できる確率を上げることが出来るが，人間にとっても難解な問題になってしまう．また，最近ではスマートフォンから文字入力するケースが増えており，PC と比較して文字入力にストレスがあるスマートフォンにおいて，ユーザビリティを低下させる１つの要因となっている．文字型 CAPTCHA のこのような問題を解決するため，画像を使った CAPTCHA が提案されている．画像型 CAPTCHA の代表的なものとして Assira[3]がある．これは人間に猫の絵を選ばせるものであるが，機械学習を使った攻撃が提案されている[4].

画像型 CAPTCHA のこれらの問題を解決するために，本論文では人間が持つ高度な知覚的補完能力に注目した．人間は断片的な情報を知覚的に補完し意味のある情報へと復元することが出来るが，これは機械にとっては難しい．この人間特有の知覚的補完能力を利用して，複数の円画像間の関係性を問う CAPTCHA を構築した．構築した CAPTCHA が機械によって推測可能か確かめるために画像解析プログラムを作成して評価を行った．併せて人間が解いた際の正答率と解答時間を計測して考察を行った．

2. 関連研究

画像を使用した CAPTCHA はこれまでに幾つか提案されている．

文献[5]は，1 枚の画像を回転させて直立させる What’up CAPTCHA である(図 1 参照)．左の列の画像が問題画像，右の列が正解画像である．A の画像は人間と機械にとって解くのが容易な問題である．人の顔が映っているため機械がそれを認識して直立させることが出来る．B の画像は鳥

† (株)KDDI 研究所
KDDI R&D Laboratories Inc.

Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

が映っているために人間にとってはこの画像を解くのは容易であるが，複数のオブジェクトが混じっているため機械には解くのが難しい問題である．C の画像は人間にとっても機械にとっても解くのが難しい問題である．

この CAPTCHA の課題は，CAPTCHA の問題を自動生成出来ない点である．画像収集した画像には図 1 左の C の画像のように人間にとって分かりづらいもの，図 1 右のギターのように正解角度が人によって分かれるものがあるためこれらを出題画像から排除しなければならない．人間にとって答えが一意に定まるものでも，一度ユーザに出題してから最も解答が多かった角度を正解角度にする必要がある．文献[5]では一度ユーザに出題し，回答結果を分析することで出題画像として登録または排除している．一度ユーザに出題しその分析結果を元にデータベースに登録するか決定するという方法は，CAPTCHA を実運用していく上で理想的ではない．文献[5]に限らず，多くの CAPTCHA にとってユーザを介さない問題の自動生成は 1 つの大きな課題である．

図 1.　What’s Up CAPTCHA?
Figure 1: What’s Up CAPTCHA?

文献[6]は抜け落ちたパズル画像を当てはめる Capy CAPTCHA である(図 2 左画像参照)．この CAPTCHA の問題では画像の 1 部分をパズルピースに置き換えているため，画素同士の色と形の繋がりが不自然になる．そのため正解画像以外では正解画像と比べて JPEG 圧縮した際のサイズが大きくなる．文献[7]が提案する解読方法では，パズルピースを順番に当てはめていきその画像をキャプチャして JPEG 圧縮する．圧縮した画像の中でサイズが一番小さいものを正解としている．実験によって 61.5%の確率で CAPTCHA が解けたことが明らかになっている．

文献[8] は歪んだ画像を回転させて元に戻す SLIDING CAPTCHA である(図 2 右画像参照)．歪曲した際にエッジの長さが元画像よりも増加する特性を利用し，それぞれの問題画像のうちもっともエッジが短くなる画像を探すというアプローチで機械にも解くことが出来る．

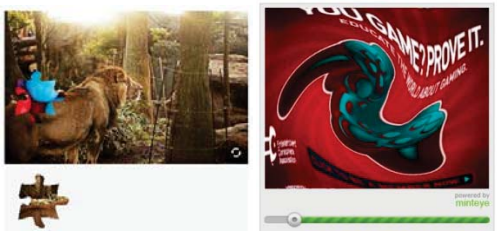

図 2. Capy CAPTCHA, SLIDING CAPTCHA
Figure 2. Capy CAPTCHA, SLIDING CAPTCHA

3. 提案方式

関連研究で述べたように, 従来の画像型 CAPTCHA では，問題の自動生成が困難である，機械的なアルゴリズムにより解けてしまう問題が作成される可能性がある，といった課題が存在する．提案方式では，これらの課題を解決することを目的としている．

3.1 概要

本研究では人間が持つ高度な知覚的補完能力に注目した．人間は欠落した不完全な図形や文字を見る場合に，視覚特性上，ある条件下ではその内容が知覚出来る．この現象を知覚心理学の分野では, アモーダル補完と呼んでいる．この人間特有の高度な知覚的補完能力は，人間が普段生活している中で頻繁に使用されている．例えば，人間が一部遮蔽物に覆われた車を見かけたとする．一部遮蔽物に覆われた物体であっても，人間はそれを車であると知覚しそれが車であることを疑わない．これは人間の知覚的補完能力によるものである．本研究ではこの能力を CAPTCHA に適用する．人間は断片的な情報を知覚的に補完し意味のある情報へと復元することが出来るが，これは機械にとっては難しいと推測される．本システムでは，ユーザに複数の円画像を表示し正しい回転角度を問うことで，画像間の繋がりが理解できているかどうか確認する．極端に画像の情報量が少ないもの, 画像間の繋がりが薄いもの(人間にとっても分かりづらいもの)に関しては, 出題されないように調整を行う．機械による隣接する 2 つの円画像のピクセル相関計算攻撃に対応するため，予め画像解析を行いそれによって正解角度を導けた問題については出題しない．

本稿で提案する CAPTCHA は図 3 のように円画像が複数出題される．このシステムでは元画像を収集し，円画像に加工してからユーザに出題するが，この過程は全てプログラムによって自動で行われる．文献[5]の課題は，正解角度を決定するため，不適切な問題を排除するために一度ユーザに出題する必要があることであった．本方式では複数枚の円画像を作成する時に正解角度を一意に定めることが出来る．また，出題画像として不適切な画像についてもプログラムによる画像解析を行うことで自動的に排除することが出来る．本稿で提案する CAPTCHA では従来の課題であ

Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

った問題の自動生成を可能にしている．

従来の CAPTCHA のもう 1 つの課題は機械的なアルゴリズムにより解けてしまう問題が作成される可能性がある点である．文献[6]が提案する CAPTCHA では画像の 1 部分をパズルピースに置き換えているため，画素同士の色と形の繋がりが不自然になる．この画像の特性を利用することで機械にも答えが特定出来る．同様に文献[8]の提案する CAPTCHA では，画像を歪曲した際にエッジの長さが元画像よりも増加するため，この特性を利用することで機械にも答えが特定出来る．本稿で提案する CAPTCHA では画像解析によって CAPTCHA が解けないようにするために，予め問題生成時に画像解析を行い機械によって解けそうなものを排除することで従来の CAPTCHA の課題を解決している．

図 3.　提案する CAPTCHA
Figure 3: The Proposed CAPTCHA

3.2 実装方法

本節では CAPTCHA の実装方法について述べる．本システムは以下の 5 つのコンポーネントから構成される(図 4).

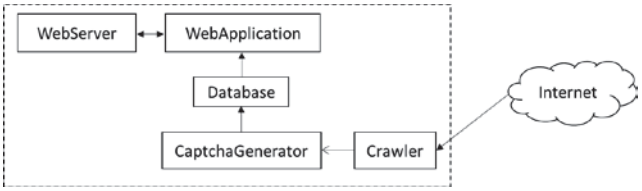

図 4. 実装方法
Figure 4: Implementation

3.2.1 Crawler

出題画像となる円画像の元画像を収集するためのクローラである．試作したシステムでは，米国の画像投稿サイト Flickr[9]の公開 API を使用して画像収集を行った．{”animal”, ”baby”, ”baseball”, ”building”, ”car”, ”dinner”, ”earth”, “girl”, “landscape”, ”sea”}の 10 カテゴリを指定している．実装は Java で行い，収集した画像は 3 万 5 千枚程度である．

3.2.2 CaptchaGenerator

1 枚の画像から複数枚の円画像を生成し，元画像と円画像をデータベースに格納するためのプログラムである．実装は Java で行い，Runnable Jar の形で出力する．円画像を効率良く生成するためにスクリプトによって 4 スレッド並列処理を行う．スクリプト言語は Ruby で実装した．画像生成アルゴリズムについては 4 章で述べる．

3.2.3 Database

円画像を保存するためのデータベースである．最も汎用的な DB である MySQL を採用した．なお，DB の操作を容易にするためにデータベース接続クライアントツールである phpMyAdmin を使用した．

3.2.4 Web Application

データベースに格納している画像を表示するための Web アプリケーションである．Web アプリケーションのサーバには汎用的である Apache Tomcat を使用した．データベースに登録されている画像データを取得して Web サーバに渡す．war ファイルを Apache Tomcat サーバにデプロイして使用する．

3.2.5 Web Server

Web サーバは Apache を採用した．Apache と Tomcat の通信は Ajax(Asynchronous JavaScript + XML)で行い，データ形式は JSON(JavaScript Object Notation)である．ユーザインターフェースの実装は HTML,CSS, JavaScript である．本サーバは幅広いユーザに利用してもらうことを前提に，グルーバル IP とドメイン(付録 B 参照)を与えている．推奨ブラウザは Firefox および Google Chrome である．

4. 円画像生成アルゴリズム

4.1 特徴点検索

人間が CAPTCHA を解く際に画像の特徴量が全く無い点に円画像が配置されると解きづらいため，予め画像の特徴量を計算し特徴点が密集している点(ノード)を各円画像の中心点候補として設定する．画像の特徴点を抽出するために OpenCV[10]の SIFT アルゴリズム[11]を使用して，各ノードに存在するキーポイントの特徴量を求める(図 5 上段中図)．SIFT では，4*4=16 ブロックに各 8 方向のヒストグラムを作成し，4*4*8=128 次元の特徴ベクトルとしてキーポイントの特徴を記述している．128 次元の各特徴ベクトルの長さはベクトルの総和で正規化されている．今回はキーポイントにおけるレスポンス値(最も強いキーポイントが選択された時の応答)を用いてキーポイントの密集している点を探索する．SIFT のレスポンス値の集合を次式で表

Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

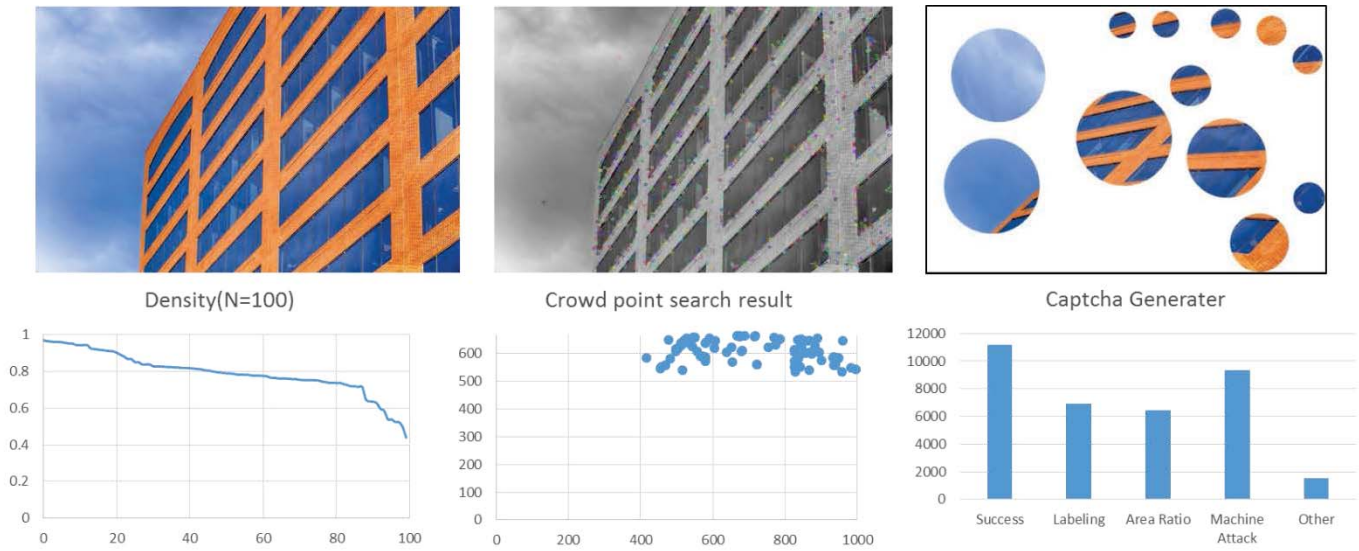

図 5. 円画像生成方法(上段左図:元画像，上段中図:元画像の特徴点抽出結果，上段右図:円画像生成結果
下段左図:密集度(N=100)，下段中図:密集点探索結果，下段右図:円画像生成結果)

Figure 5. Circle image generation method.

す．

$$\mathcal{R} = \{r_j \mid j = 1 \cdots \mathcal{N}\} \tag{1}$$

特徴量の密集点を探索するために，ノード$i(x_i, y_i)$とノード$j(x_j, y_j)$のユークリッド距離に比例してスコアが低くなるように設定する．ノード$i(x_i, y_i)$とノード$j(x_j, y_j)$のユークリッド距離$d(i,j)$は次式により算出する．

$$d(i,j) = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \tag{2}$$

よりキーポイントの特徴量が高く密集しているノードを選択する為に式(1)，式(2)を使用してノード$i(x_i, y_i)$における特徴点の密集度(Density)$\mathcal{D}_i$を次式で定義する．

$$\mathcal{D}_i = \sum_{j=1}^{\mathcal{N}} \frac{r_j}{d(i,j)+1} \tag{3}$$

$\mathcal{N}$を 100 として計算を行った(図 5 下段左図)．ここで得られた密集度の高いノードを XY 軸にプロットしたものを示す(図 5 下段中図)．ここで求めた密集度の高いノードを中心点候補として円画像生成を試みる．以下にその手順を示す．

Step1.　1 枚目の円画像の中心点を求める．画像の中心を中心点としある一定の半径をとるようにする．

Step2.　2 枚目以降の円画像の中心点に関しては，図 5 下段右図で示した点を中心点候補とする．中心点の選定に当たり，中心点同士がある一定の距離を保っていない中心点候補は排除する．ここで自然数$\mathcal{N}$回分円画像生成試行を繰り返した結果，円画像枚数が規定値(5 枚)を超えなかった場合には，ランダムな点を円画像の中心点候補とする．円画像生成試行の上限(100,000 回)に達するか，円画像の最大枚数(12 枚)に達した時点で終了する．

Step3.　Step2 で設定した中心点に設置する各円画像の半径を求める．円画像の半径に関しては，基本的には円の中心点から四方向の壁までの最小距離を設定するが，他の円画像と重なる場合には重ならないように円画像の半径を縮小する．

4.2 円画像の配置確認

生成した円画像の配置場所が図 6 左のような場合，円同士の関係性が分からないため人間にとって解答するのが難しい．一方，図 6 右のような場合であれば円同士の関係性を推測することが出来る．4.1 節で生成した円画像が輪郭線上に 3 つ以上の円が配置されるように画像処理を行う．まず画像から輪郭線を自動的に抽出し，そのあとで抽出した線に色づけ(ラベリング)を行う．輪郭線，ラベリングの処理については次節以降述べる．

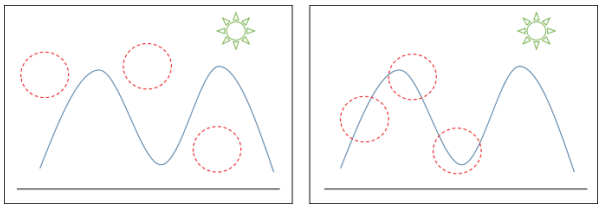

図 6. 円画像の配置場所

Figure 6. Location of circle

4.3 輪郭線抽出

画像の輪郭線の検出には一次微分を用いた Sobel(ゾーベル)フィルタを用いる[12]．Sobel フィルタを使ったエッジ

Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

検出では，式(4)で示した垂直方向エッジ検出オペレータと式(5)で示した水平方向エッジ検出オペレータを使用して，検出結果を足し合わせることで求める．

$$\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix},\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix} \quad (4)$$

$$\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix},\begin{bmatrix}1 & 2 & -1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} \quad (5)$$

輪郭線抽出結果を図 7 左に示す．十分に輪郭線を検出出来なかった画像は出題画像として不適切なため，2 値画像の白黒の割合(輪郭線を黒，その他を白としたときの黒の割合)が 4%よりも低いものは排除する．

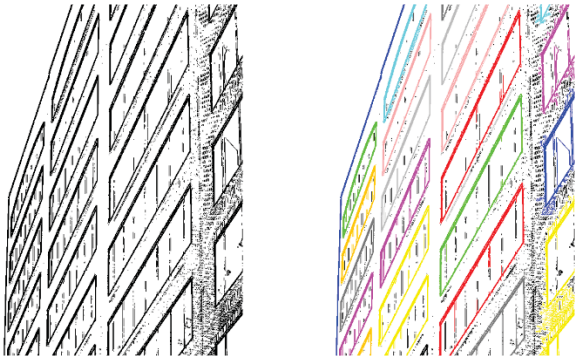

図 7．画像処理結果

Figure7. Image Processing Result

4.4 ラベリング

4.2 節で述べた輪郭線の抽出を行ったあとに，ラベリングによって各輪郭線にラベル付けする．ラベル付けとは 2 値画像内のある点の 8 近傍(縦，横，斜め方向)を調べ黒の部分が連続した画素に同じ番号(ラベル)を割り振る 処理のことである[13].

元画像に対して輪郭線をラベリングした結果を図 7 右に示す．この際に輪郭線が一定数以上抽出出来なかった画像は出題しない．また，各円画像の円弧上に含まれているラベル番号を数え，最も多く数えられたラベル番号が 2 つ以下だった場合も出題しない．この処理を行うことで，1 つの輪郭線に最低 3 つの円画像が配置されることになり，人間にとって画像間の繋がりを推測するのが容易になる．

4.5 円画像生成結果

図 5 下段右図に円画像を生成した結果を示す．縦軸に画像枚数を示し，横軸は画像生成結果である．Success は画像生成に成功した(CAPTCHA の出題画像となる)画像の枚数である．Labeling はラベリング処理を行う際に，輪郭線の数が少なかったもの，ラベル上に含まれる円の数の最大値が 2 つ以下だった画像の合計枚数である．Area Ratio は輪郭線を抽出した際に，2 値画像の白黒の割合が低かった画像の枚数である．Machine Attack は評価の際に使用したプログラムによって解けた画像の枚数である．これについては 5 章で詳しく述べる．Other は画像サイズが小さかった画像の枚数である．3 万 5 千枚程度の画像に対して CAPTCHA 生成プログラムを実行し，円画像の生成に成功した画像は 1 万枚程度である．CAPTCHA 生成プログラムの実行時間は 1 枚につき 1000~2000(ms)程度である．実行時は 4 スレッド並列処理を行った．

表 1．実行環境

OS	Ubuntu14.04.01 LTS
CPU	Intel Core i7 3770K 3.50GHz * 8
Memory	16GB

5. 画像解析

画像を使用した CAPTCHA の 1 つの課題として，文献[6]や文献[8]のように画像解析によって CAPTCHA が突破される可能性がある．本論文では完全なチューリングテストを作成するために，作成した CAPTCHA に対して画像解析を行い解けた問題は予め除外することで画像解析攻撃に対して耐性をもたせる．

本節では作成した CAPTCHA に対して画像解析し CAPTCHA を解く方法を述べる．2 つの円を選択し円弧上の点のピクセル同士の相関を調べその相関値が一番低かった角度を正解角度として推定する．次節で詳細のアルゴリズムを説明する．

5.1 画像解析アルゴリズム

本節では CAPTCHA を解くための画像解析アルゴリズムを述べる．比較対象円を I，被比較対象円を U とし，半径をそれぞれ，I_r, U_rとする．

Step1 : I と U の中心点を結んだ際の x 軸から見た角度I_θ, U_θをそれぞれ求める．

2 つの円の中心をそれぞれI_c(x1,y1)，U_c(x2,y2)とし，

$$dx = x2 - x1 \quad (6)$$

$$dy = y2 - y1 \quad (7)$$

とすると，

$$I_\theta = \tan^{-1}(dx, dy) \quad (-\pi \ll \tan^{-1}(dx, dy) < \pi) \quad (8)$$

$$U_\theta = \alpha + 180 \quad (9)$$

で表すことが出来る．

Step2 : I と U の円弧上の点I_p, U_pを求める．

調査角度 dS，回転角度 dR を以下のように定める．

$$-90 \le dS \le 90 \quad (10)$$

$$0 \le dR \le 360 \quad (11)$$

円弧上の点は，

Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

$$I_p = (I_r \cos(I_\theta + dS + dR), I_r \sin(I_\theta + dS + dR)) \quad (12)$$

$$U_p = (U_r \cos(U_\theta + dS + dR), U_r \sin(U_\theta - dS + dR)) \quad (13)$$

となる．

Step3：円弧上の点の距離を求める．
点I_p, 点U_p間の距離を L と定義する．

$$L = \begin{cases} l & (dS = 0) \\ l + divL & (dS \neq 0) \end{cases} \quad (14)$$

$$l = \sqrt{\{I_r \cos(I_\theta) - U_r \cos(U_\theta)\}^2 + \{I_r \sin(I_\theta) - U_r \sin(U_\theta)\}^2} \quad (15)$$

$$divL = (I_r + U_r)(1 - \cos(dS)) \quad (16)$$

Step4：円弧上の 2 点の RGB 三原色の二乗誤差(Square Error)を求める．二乗誤差の算出の際に，距離 L がある一定の距離MAX_Lを超える場合には計算を行わない．W は画像の横サイズ，H は画像の縦サイズとする．

$$MAX_L = \frac{\sqrt{W^2 + H^2}}{10} \quad (17)$$

点I_p点U_pの RGB の二乗誤差を$d(I_p, U_p)$で表し，
回転角度 dR において dS を-90 度から 90 度まで動かした時の Score を以下のように定義する．

$$Score[dR] = \sum_{dS=-90}^{90} \frac{d(I_p, U_p)}{L^2} \quad (18)$$

この関数を 0~360 度の全角度で試行するため，円画像数を n とすると全計算量$\log(n)$は，

$$\log(n) = n(n-1) \sum_{dR=0}^{360} Score\,[dR] \quad (5 \leq n \leq 12) \quad (19)$$

となる．

5.2 画像解析結果

5.1 節で示した式(18)の Score を 0 から 360 度の範囲で算出した結果が図 8 である．Score が最も小さくなる時の角度を機械による推定角度とする．推定角度と正解角度が一致した時に機械によって解けたものとする．

図 8. 各回転角度におけるスコア
Figure 8. Score at each rotate degree

図 8 の上の図は機械によって角度が推定出来なかった例，図 8 の下の図は機械によって正解角度が推定出来た例である．機械による正答率は 26.3%程度(9341/35398)であった．機械によって解けた問題は CAPTCHA の出題画像から除外した．

6. ユーザ実験

提案方式の有効性を確認するために，利用者を想定した被験者により評価を実施した．本節ではその評価実験の手順と結果を説明する．

6.1 実験手順

1 人につき CAPTCHA を 100 問出題し解いてもらう．被験者は社内の 10 名で，年齢層は 20 代から 50 代である．CAPTCHA の正答率と解答時間を計測し分析を行った．実験を行うにあたり，ユーザが CAPTCHA のページにアクセスした際にブラウザに Cookie を設定しランダムにユーザ ID を付与する．Cookie のライフタイムはブラウザのプロセスが終了するまでである．Cookie 内にユーザ ID を仕込むことでユーザ毎の正答率を求める．解答時間の計測に関してはブラウザで CAPTCHA 画面にアクセスした時点から解答ボタンを押したまでの時間を計測する．Ajax の通信時間，円画像の描画時間に関しては軽微なものとみなす．正解角度と円画像の回転角度を 12 度刻みに設定し，正解許容角度は 0 度で実験を行った．正答率と解答時間をデータベースに保存し分析する．

Vol.2015-DPS-162 No.53
Vol.2015-CSEC-68 No.53
2015/3/6

6.2 実験結果

正答率と解答時間の結果と考察を述べる．

6.2.1 正答率

ユーザ 10 人の平均正答率は 91.9%であった．ユーザ別の正答率を表 2 に示す．ほとんどユーザが 9 割以上の正答率を示しており十分に実用的であることが分かった．文献[5]のユーザ実験結果と比較すると，許容角度 12 度，画像数が 1 枚から 6 枚に増えるにつれて正答率は 93.10(%), 86.68(%), 80.70(%), 75.13(%), 69.94(%), 65.12(%)と低下しており，100 問出題してこの正答率は高いと言える．

表 2. 実験結果

User1	88%	User6	98%
User2	91%	User7	95%
User3	98%	User8	95%
User4	91%	User9	93%
User5	92%	User10	78%
		Average	91.9%

不正解だった問題をベースとなった写真のカテゴリ別に分類したものが図 9 のグラフである．このカテゴリとは 3.3.1 節で述べたクローラで集めてくる際のキーワードである．不正解が突出して多かったカテゴリは earth(18 回), animal(17 回)であった． 小動物や地球の被写体が中心に寄っていてかつ小さい場合には中心円に全ての輪郭が含まれてしまい，他の円がユーザへのヒントにならずに画像間の意味を補完することが出来なかったことが主な原因として考えられる．これらの不正解が多かったカテゴリを出題画像から排除することで正答率を更に向上させることが出来ると考えられる．

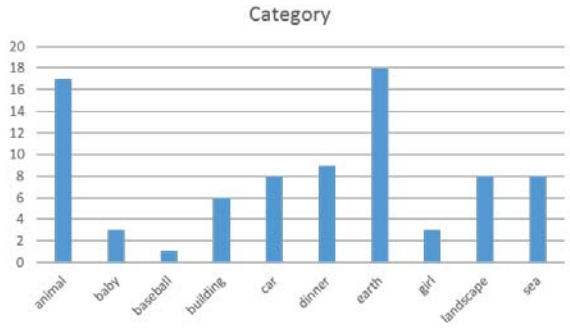

図 9. カテゴリ

Figure 9. Category

6.2.2 解答時間

ユーザの解答時間の統計結果を図 10 に示す．解答時間の四分位数(データを値の大きさの順に並べたときに，4 等分する位置にくる値)を求めグラフ化したものである．なお，外れ値は 60,000(ms)とした．それぞれの被験者の中央値を比較すると，最も解答時間が短い被験者で 5722.5(ms)，最も解答時間が長い被験者で 11082(ms)であり，大半のユーザが 10 秒かからず解いていることが分かる．

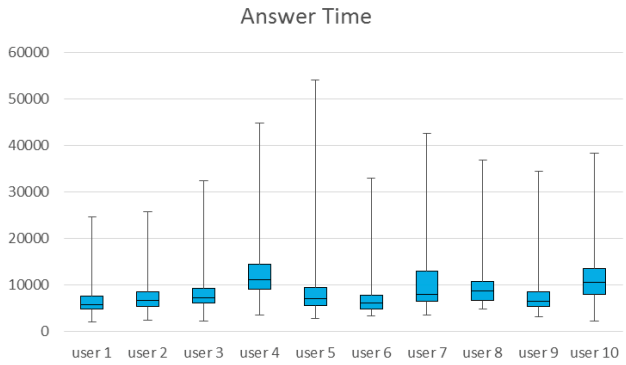

図 10. 解答時間

Figure10. Answer Time

7. 考察

7.1 Brute Force Attack

機械が無作為に Captcha の正解角度を解答する総当たり攻撃に対する耐性について考える．機械が無作為に解答角度を選択した場合の確率は二項分布の一般式で計算される．成功数 x，試行回数 n，成功確率 p とすると，

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x} \tag{20}$$

ここで確率 p は問題の正解角度を何度刻みにするかに依存する．正解角度間隔をθ=12,4,1 とすると，確率 p はそれぞれ p=1/30,1/90,1/360 となる．n=100 としてそれぞれの成功確率を図 11 に示す．θ=1 の時の機械の総当たり攻撃による成功数をそれぞれ求めると，$p(3) = 0.061228$, $p(4) = 0.015222$, $p(5) = 0.003019$ である．つまり，機械がランダムで 100 回解答した時に Captcha を 3 回解ける確率は 6.1%，4 回解ける確率は 1.5%，5 回解ける確率は 0.3%である．よって，総当たり攻撃によって問題が解ける確率を 1%以下にするには，正解角度間隔を 1 度，出題回数は 5 回が望ましい．

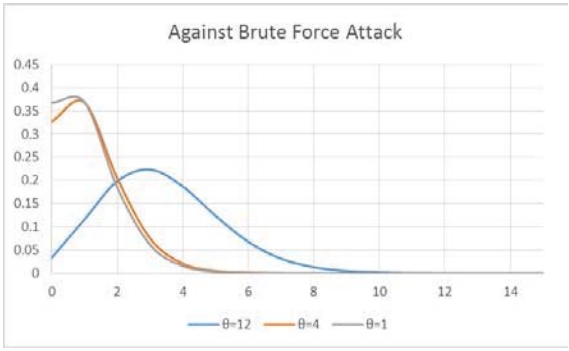

図 11. 二項分布(θ=12, 4, 1, n=100)

Figure 11. Binomial Distribution (θ=12, 4, 1, n=100)

7.2 DeepLearning

DeepLearning を使った画像分類に関する研究が行われている．DeepLearning とは最近注目されている画像認識技術である．それまでの一般物体認識では入力画像から SIFT などで特徴量を抽出しベクトル化した後，それを SVM(Support Vector Machine)などで分類するのが主流であったが，DeepLearning の登場によって衰退した．DeepLearning は教師なしニューラルネットワークを繋げて多層にしたものである．文献 [14]では，YouTube の動画からランダムの画像を 1000 万枚取り出し DeepLearning を行った．その結果，人間の顔，猫の顔に反応するニューロンを作成出来たことが報告されている．今後この DeepLearning を使用した画像認識の向上によって，本 CAPTCHA への新たな脅威となる可能性がある．

7.3 SmartPhone 対応

本 CAPTCHA はスマートフォンにも対応している．スマートフォンでの操作は指でのタッチ操作になるためタッチイベントを取得し，指でなぞった距離に比例して円画像を回転させることが出来る．

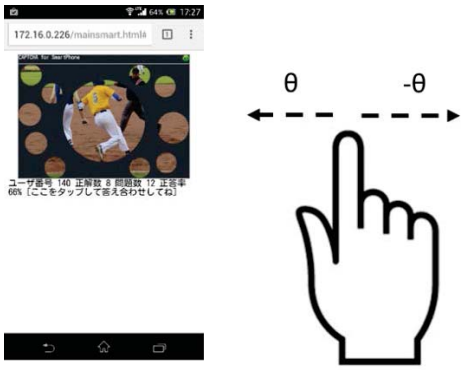

図 12. スマートフォンでの操作画面
Figure12. Operation screen of SmartPhone

8. 終わりに

本稿では，人間特有の知覚的補完能力を利用して複数の円画像間の関係性を問う CAPTCHA を作成した．従来の画像型 CAPTCHA の欠点であった画像解析に対して耐性をもたせるために，画像解析を行い機械にも正解角度が推測できる問題は除外した．また，人間にも解き易いように円画像間同士の繋がりが推測し易いように CAPTCHA を作成した．評価においては十分な正答率と許容出来る解答時間が得られており実用可能であることが分かった．今後はこのシステムを公開し，他のサイトで幅広く使用してもらう予定である．

謝辞 実験にご協力頂いた皆様に，謹んで感謝の意を表する．

参考文献

1) The Official CAPTCHA Site,
http://www.CAPTCHA.net/
2) Yan, j. and Ahmad, A.S.E.: Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms, 2007 Computer Security Applications Conference, pp.279-291 (2007).
3) Jeremy Elson,John R. Douceur.Jon Howell, and Jared Saul: Asirra: A CAPTCHA that Exploits interest-Aligned Manual Image Categorization, Proceedings of 14th ACM Conference on Computer and Communications Security(CCS)(2007).
4) Philippe Golle:Machine learning attacks against the Asirra CAPTCHA, Proceedings of the 15th ACM Conference on Computer and Communications Security(CCS), pp.535-542(2008).
5) Grossweiler R, Maryam K and Baluja S:What's up CAPTCHA?:a CAPTCHA based on image orientation, WWW '09 Proceedings of the 18th international conference on World wide web, pp.841-850(2009).
6) Capy CAPTCHA,
https://www.capy.me/jp/
7) Carlos J.H.C, Maria D.R.M and David F.B :Side-channel attack against the Capy, HIP 2014 International Conference on Emerging Security Technologies.
8) minteye,
http://www.minteye.com/
9) Flickr,
https://www.flickr.com/
10) OpenCV,
http://opencv.org/
11) 藤吉弘亘:Gradient ベースの特徴抽出 -SIFT と HOG-,
電子情報通信学会信学技報 Vol.107 No.207.
12) CodeZine,
http://codezine.jp/article/detail/129
13) 画像処理ソリューション,
http://imagingsolution.blog107.fc2.com/blog-entry-193.html
14) Quoc V. Le et al.: Building High-level Features Using Large Scale Unsupervised Learning, Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.

付録

付録 A CAPTCHA URL

PC : http://captcha.kddisec.jp/main.html

SmartPhone : http://captcha.kddisec.jp/mainsmart.html