

# 特徴点を自動生成する生体認証システムの マルチプラットフォーム化

山崎尚人<sup>†1</sup> 堂園浩<sup>†2</sup> 中頭達也<sup>†3</sup>

近年の情報化社会の発達により、情報端末が一般的なユーザまで普及したことは疑いようのない事実である。しかし、情報を守るという意識は端末所持者には浸透しきれず、個人情報等の重要な情報が危険にさらされていることが多い。本研究は、筆跡と筆速による生体認証を行うユーザビリティの高い個人端末向けの認証システムの実現を目的とする。

## Biometric Authentication System Which Generates Feature Points Automatically Using HTML5

NATOTO YAMASAKI<sup>†1</sup> HIROSHI DOUZONO<sup>†1</sup>  
TATSUYA TYUTOU<sup>†2</sup>

Recently, the smartphone and tablet devices are in wide spread use with the growth of electronic technology. These devices keep the personal information, and are used to access secure system. Recently, the smartphone and tablet devices are in wide spread use with the growth of electronic technology. These devices keep the personal information, and are used to access secure system.

### 1. はじめに

近年、電子技術の発達により、情報の電子化が進み、スマートフォンやタブレット端末を所持する人が増加している。情報端末ではクラウドサービスの使用により膨大な量のデータのやり取りや、ソーシャルネットワークサービスの使用のため個人情報等の重要な情報が保存されており、インターネット上の情報アクセスにも使用される。そこで問題となってくるのが情報端末の所持者が大切な情報を守るという意識がどれほどあるのかという点である。現在の情報化社会の中で個人情報を盗まれるという危険性が十分に認識されているとは言いきれない。今後さらなる技術の発展により、利便性が重視されセキュリティが疎かになることが考えられる。そこで、より柔軟性が高く利用しやすいセキュリティシステムとして、[1]のiOS向けの筆跡と筆速から特徴点を自動的に生成する生体認証システムが提案された。この認証システムは、タッチパネル状に描かれた手書き図形を用いて認証を行う方式のひとつであるが、図形の登録時に特徴点を抽出し、ユーザビリティの向上や認証に用いるパラメータの削減を行っている。

ここ数年、AJAX やWEBアプリなど、端末上からクラウド上のアプリケーションを実行したり、データを参照するシステムが増えてきている。特に携帯端末上ではFIREFOX OS や Chromo OS など、クラウド上におかれたア

プリケーションを実行するOSも出現している。そこで[1]のシステムを拡張して、ブラウザを使用できるものなら使用が可能といわれるHTML5上に実装する方法について検討を行った。また、同様にマルチプラットフォームで実行可能なHSP上に実装し、認証精度などの確認を行った。

### 2. 使用言語

#### 2.1 HTML5

HTML:HyperText Markup LanguageとはWeb上のドキュメントの記述に用いられるマークアップ言語で、HTML5からアニメーションの制御、データの保存、ソケット通信などが定義されており、以前より開発できるシステムの幅が広がっている。HTML5は用いることでブラウザを使用することができる端末でシステムを動かすことが可能になる。その点で将来性のある言語だと考えられる。

#### 2.2 Javascript

Mozillaが仕様の策定をして実装しているオブジェクト指向のスクリプト言語である。動的なWebサイト構築でWebアプリケーション等のユーザインタフェースの開発に用いられる。

#### 2.3 HSP3 with HSP3dish

HSPはwindows上で動作するスクリプト言語である。文法が簡潔で、多彩な命令セットがあり比較的簡単にアプリケーションソフトを作ることができる。また、HSP3DISHを用いることで、AndroidやiOSなどマルチプラットフォームに対応することができる。

本研究では、HSP3を用いた認証システムの認証、およ

<sup>†1</sup> 佐賀大学工学系研究科先端融合工学専攻

<sup>†2</sup> 佐賀大学理工学部電気電子工学科

び、HSP3dish を用いたマルチプラットフォームでの実装を行い、認証精度の確認を行った。

### 3. 自動的に特徴点を生成するシステム

本研究では認証にタッチパネル上にかかれた図形の知識属性と生体的特徴量を用いる。このような図形を用いた生体認証を行う際、どの特徴量を用いる場合でも完全なフリーハンド曲線から認証を行うことは非常に難しい。理由は自由に曲線を描画できてしまうと利用者が毎回同じ図形を再現することが難しいからである。[2]において、画面上にあらかじめ描画された図形をなぞることによる認証方式が提案されている。この方式では、知識属性を用いないことで図形の忘却や同一性の問題を解決することを目指しているが、その反面、十分な認証精度が得られているとは言い難い。そこで本研究では、筆跡による認証を行う場合、ある程度の制限を設けて元の筆跡に近いものを再現する方式を用いるものとする。実際には、図形の登録時に特徴点を生成し、認証時には特徴点のみを表示し、その特徴点を参考に元の図形に近い曲線を再現するものである。本方式に近いものとして、はアンドロイド端末で用いられているロック解除画面(図1)があるが、本方式は、(1)図形の自由度が高い、(2)知識認証のみでなく生体認証も行うことを特徴とする。



図1 アンドロイドのロック解除画面  
 Figure 1 unlock screen of android

#### 3.1 特徴点の自動取得

描画された筆跡から特徴のある点を自動的に探し出すシステムである。これを筆跡再現の補助に用いることで利用者がストレスを感じることなく自分の描いた曲線を再現しやすくなり、再現率が高くなれば特徴量も安定し、本人が認証を失敗する確率(FRR)を低下することができ、認証システムとしての質を上げることができると考える。特徴点の取得画面を図2左に示す。

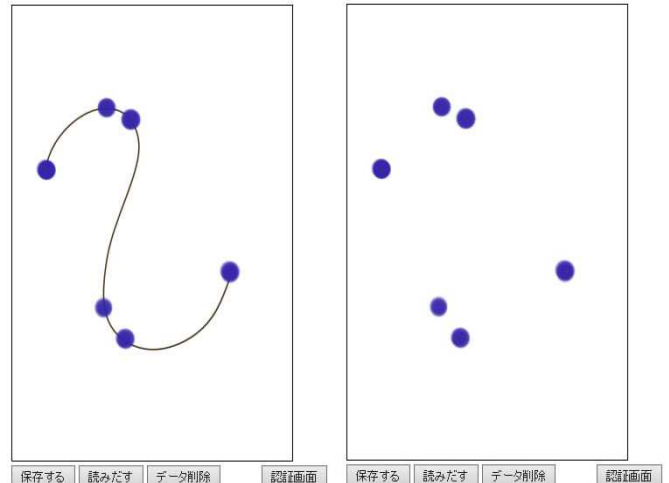


図2.特徴点取得画面と認証画面

Figure 2 Feature acquisition screen and Attestation screen.

#### 3.2 特徴点の取得方法

特徴点は描画された曲線上の座標から計算する。定間隔で三点を取得し、この三点から曲線の内積を用いることで曲がり角度の大きい点を特徴の大きい点としてマーキングを行う。今回  $\cosine \theta$  の値が-0.95 より大きい場合マーキングを行った。これより小さくしていくと緩やかな曲線部分をマーキングできないことが確認された。しかし、このままでは特徴点の数が多くなりすぎるためマーキングされた点から、三点を取得し一点ずつずらしながら再び内積より角度を求め  $\cosine \theta$  の値が-0.95 以上となる時の座標を特徴点とする。さらに精度を上げるために曲率。曲率の計算は

$$d(t) = \frac{y(t+1) - y(t)}{x(t+1) - x(t)}$$

$$\text{曲率} = \left( \frac{1}{(1 + (d(t))^2)^{\frac{3}{2}}} * (d(t+1) - d(t)) \right)$$

として計算を行っているが、描写された曲線が垂直に近くなり X 座標の変化量が小さくなってしまうとわずかな手ブレ等でも特徴点としてマーキングを行ってしまう。その場合認証オブジェクトが増えすぎて生体認証を行う事が出来ない。そこで対策として曲率の計算を行う際に X 座標と Y 座標の値を入れ替えた計算も行い、それぞれが 0.2 以上の曲率を得る事でマーキングが可能となるようにアルゴリズムを改良した。

結果として同じような曲線を描写した場合の手ブレによる誤認識は改善された。

#### 3.3 認証オブジェクトの重複

新たな問題として誤認識ではないが、マーキング点が近す

ぎて重なってしまう例が現れた。この問題に対しては重なった点の曲率を参考に新たな統合点を作成してオブジェクトとする事で対策を行った。統合点は元のマーキング点を持つ曲率の値を比較して、その比率に応じてマーキングを行う(図3)。

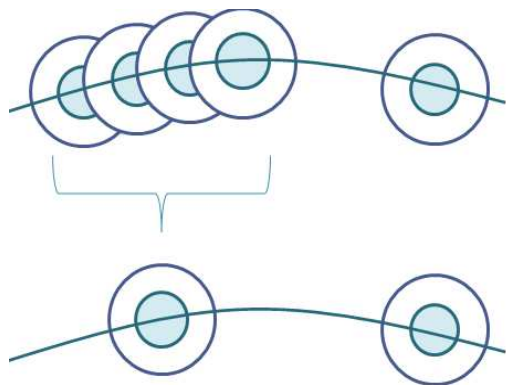


図3.点の結合

Figure 3 Combination of feature points.

### 3.4 認証方式

前述のように本方式では、知識属性を用いた認証と生体認証を行う。知識属性を用いた認証においては、図2右のような特徴点のみを表示し、この特徴点間を登録した図形順になぞるものである。実用上は特徴点が少ない、単純な図形の場合は簡単に再現されてしまう可能性があるが、ある程度複雑な図形の場合、再現は困難であると考えられ、このことに関しては、後に実験において確認するものとする。生体認証には筆速を用いるものとし、図4に示すように特徴点を利用して作成した認証点同士を結合する際に取得する。特徴点A,Bを中心とした円を通過した時間を $t_1$ ,  $t_2$ とすると、 $t_2-t_1$ を保存値と比較する。これをBC間、CD間と繰り返し、全セクションにおいて認証値と保存値の誤差が点の間の距離に応じて設けられている閾値の範囲内の時認証成功となる。

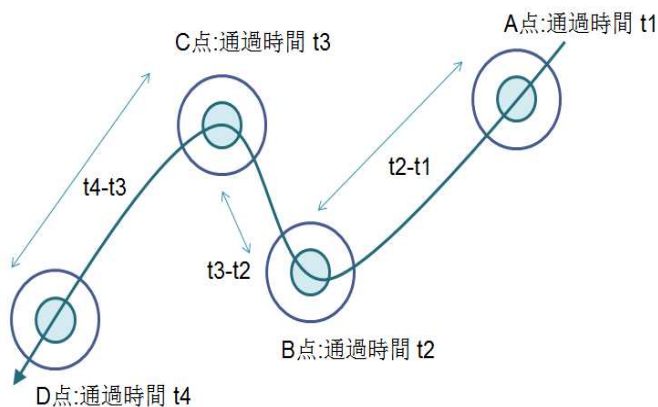


図4.筆速の取得

Figure 4 Acquisition of the speed which draws a line.

## 4. アプリケーションの構成

### 4.1 HTML5 上での特徴点生成システム

後述のようにHTML5上で認証まで行うのは問題があるため、HTML5では特徴点生成システムのみの実装を行った。実際、図1の画面はHTML5上での実行結果となっている。

### 4.2 HSP 上での認証実験システム

実装のしやすさから、HSP上に認証実験を行うためのシステムを実装した。実際には、特徴点の生成と保存を行い、生成した特徴点を用いて認証実験を行うシステムを作成した。実際のシステム開発はタッチパネルがついたVAIO duo 11を用いてWindows 8上で行い、タブレット端末上での実験はAsus Memopadを用いて、HSP3Dish上でプログラムを実行した。また、tablet端末上での実験では図形の登録も端末上で行った。

### 4.3 認証手順

[手順1]タッチパネル上に曲線を描写して特徴点を設定する。

[手順2]取得した特徴点を用いて認証パスを作成する。認証パスは特徴点の座標を用いた結線時の筆速を記録。

[手順3]表示された特徴点を結線し、認証を行う。

### 4.4 HSP での実行結果

HSPで開発されたシステムでの結線順を教えていない場合の結果を表1、結線順を知らせていない場合の結果を表2そしてtablet端末上でHSP3Dishを用いて動かした場合の結果を表3に示す。認証画面では曲線は表示されず、特徴点しか表示されないようになっている。セキュリティとしての強度は、表1の結果から4人に3パターンの認証を試みた結果だが本人以外は認証パスの再現はできなかった。結線開始の点を当てることも難しいことが分かった。つまり、特徴点のみから認証パスの再現はほぼ不可能だと考えられる。

結線順を知っている場合では何回かに1回認証に成功してしまう。これを回避するには特徴点の数を増やすことで防げることが表2よりわかるが、その代わりに本人拒否率も上がっていることが確認できる。Tablet端末では他人許容率は下がっているが本人拒否率は上がるという結果になった。

表 1.結線パターンを知らない場合の認証結果

Table 1 Authentication results for unknown pattern

|     | パターン 1 | パターン 2 | パターン 3 |
|-----|--------|--------|--------|
| 登録者 | 10     | 9      | 7      |
| A   | 0      | 0      | 0      |
| B   | 0      | 0      | 0      |
| C   | 0      | 0      | 0      |
| D   | 0      | 0      | 0      |

表 2.結線パターンを知らない場合の認証結果

Table 2 Authentication results for known pattern

|     | パターン 1 | パターン 2 | パターン 3 |
|-----|--------|--------|--------|
| 登録者 | 10     | 9      | 7      |
| A   | 3      | 2      | 0      |
| B   | 1      | 0      | 0      |
| C   | 1      | 2      | 0      |
| D   | 4      | 3      | 0      |

表 3.Tablet 端末の認証結果

Table 3 Authentication results using tablet device

|     | パターン 1 | パターン 2 | パターン 3 |
|-----|--------|--------|--------|
| 登録者 | 9      | 7      | 4      |
| A   | 1      | 0      | 0      |
| B   | 0      | 0      | 0      |
| C   | 1      | 1      | 0      |
| D   | 3      | 0      | 0      |

## 5. HTML5 を用いた認証システムの実装

本研究では HTML5 を用いて WEB アプリケーションなどの生体認証システムを構築することを目的としている。もっとも単純な実装方法としては、WEB アプリケーションの HTML5 のソース自体に認証システムを実装する方法が考えられる。この方法はコンピュータを全く知らない素人には有効かもしれないが、ある程度知識をもったユーザには簡単に破られてしまう。なぜなら、HTML5 はソースコードがそのままネットワークを転送されてくるため、ソースコードを見て、認証部分をスキップするようにプログラムを書き換えることが容易だからである。また、HTML5 のソースコードを隠蔽する方法も提案されているが、いずれにせよネットワークを平文で送られてくるため、何らかの方法で取得することが可能であると考えられる。

そこで、HTML5 を用いた認証システムを実装する場合、HTML5 では特徴点の抽出、結線順序の取得、筆速などの生体情報の取得のみを行い、その情報をホストシステムに暗号化を行って転送し、ホストシステムにおいて認証を行う方法が有効であると考えられる。また、認証後に実行するアプリケーションは、認証成功時のみユーザのシステム

に転送するようにする必要もある。

このようにホストシステム上で実装する方法において本方式の利点として、パラメータが少ないことがあげられる。本研究で用いられているような図形パターンなどを用いた生体認証方式においては、図形パターンをスキャンしたすべての座標データを比較する方法が用いられることが多く、[2]においても 1 図形につき数 100 から数 1000 の座標データを取得しているが、ネットワークを転送するとその負荷も大きくなるが、本方式では特徴点間の筆速のみを転送するため、大幅にデータ量を削減可能である。また、ホストシステムでの認証システムを記述するスクリプト言語としては、HSP を用いることも可能であり、この研究で作成した認証実験システムを応用することもできると考えられる。

## 6. おわりに

本研究では HTML5 上の生体認証システムとして有効であると考えられる、特徴点を自動的に生成する生体認証システムを提案し、その認証精度を HSP 上で確認した。

結果として今回作成した特徴点を使用するシステムの認証システムとしての可能性を確認できた。特に知識属性と組み合わせた場合、十分な認証精度を持つが、純粋に生体認証システムとしては FRR, FAR とともに認証精度が不十分であり、認証アルゴリズムの改良が必要であると考えられる。また、今後 HTML5 とホストシステムを用いた認証方式を実装し、その実用性と問題点を検討する必要があると考えられる。

**謝辞** ご指導いただいた教員また協力頂いた研究室の皆様に、謹んで感謝の意を表する。

### 参考文献

- [1] Hiroshi Dozono, Youki Inaba, Masanori Nakakuni, "A Biometric Authentication System That Automatically Generates Feature Points", Proceedings of the International Conference on Security and Management 2013 (2013)
- [6] Hiroshi Dozono and Masanori Nakakuni, et.al: "The Analysis of Pen Pressures of Handwritten Symbols on PDA Touch Panel using Self Organizing Maps", Proceedings of the International Conference on Security and Management 2005, pp.440-445(2005)