

# タッチパネルによる手指の行動的特徴を用いた 生体認証に関する一考察

居城 秀明<sup>1,a)</sup> 金岡 晃<sup>2,b)</sup> 岡本 栄司<sup>2,c)</sup> 金山 直樹<sup>2,d)</sup>

**概要：**タッチパネルを搭載したスマートフォンやタブレット型端末といったスマートデバイスで利用される認証方法は、パスワード認証やパターン認証に代表される”知識認証”が一般的に用いられているが、認証時の背後からの覗き見に弱く端末からの個人情報漏洩リスクが危惧されている。今日、”知識認証”にかわる認証方法として、認証時の覗き見に対して耐性を持つことで知られる”生体認証”ベースの、個人の手指の行動的特徴を用いる手法が Sae-Bae 等などによって提案されている。しかし、これらの研究ではタッチパネルから得られる手指の情報に関して、取得する座標軌跡の細かさや認証精度、認証時間についての十分な検討がなされておらず、認証システムの実現には議論の余地が残る。本稿では、手指の行動的特徴を取得する際の、タッチパネルからの座標取得頻度に着目した。各個人を識別する際の最適な座標取得頻度について認証時間・認証精度とのトレードオフを考慮し高速かつ高精度な認証を目指す。

**キーワード：**生体認証、タッチパネル、パターン認識、動的計画法、ジェスチャ

## 1. はじめに

### 1.1 背景

2007年に米 Apple 社が発表した“iPhone”を皮切りとして、多機能端末（スマートデバイス）が一般のユーザーに広く普及し始めた。Google の調査 [1] によれば 2012 年現在、多機能端末の 1 つであるスマートフォン普及率はアメリカで 44%、カナダで 33%、イギリスやオーストラリアでは 50%を上回る。

このような急速な普及に対し、第三者による不正利用に対する安全性が懸念されている [2]。スマートデバイスは通常の携帯端末に比べ、ユーザーアカウントやそのパスワードなどを記録することで、様々なサービスの個人情報へのアクセスを容易にしている。このため端末が不正に利用さ

れた場合、情報漏洩の被害はより深刻なものとなる。

そうした個人情報を保護するための手法の 1 つに個人認証があげられる。個人認証に関しては利用状況に応じて様々な手法が提案されているが、公共の場所等へ携帯することの多いスマートデバイスに相応しい認証手法が望まれている。

### 1.2 生体認証

スマートデバイスなど、モバイル端末には多くの場合、個人認証の手法としてパスワードや、複数の点のつながりによるパターンといった、本人のみが知りうる知識による認証（知識認証）が用いられている。通常、これらの認証は 4 桁程度の数字や 9 つ程度の点の結び方のパターンを用いており、背後からの覗き見によるなりすましに弱いという脆弱性を持つ。対策として長いパスワードや複雑なパターンの使用が考えられるが、これらは利便性を失うため認証システムの受容性が損なわれる。

近年モバイル端末向けの認証手法として生体情報を用いた認証（生体認証）が注目されている。その理由として、生体認証が一般に覗き見によるなりすましに耐性を持ち、かつ認証が簡便であるという点があげられる [3]。

### 1.3 タッチパネル利用の意義

現在普及しているスマートデバイスの多くはタッチパネ

<sup>1</sup> 筑波大学 大学院 システム情報工学研究科  
〒 305-8573 茨城県つくば市天王台 1-1-1  
Graduate School of System and Information Engineering,  
University of Tsukuba.

<sup>2</sup> 筑波大学 システム情報系  
〒 305-8573 茨城県つくば市天王台 1-1-1  
Faculty of Engineering, Information and Systems, University  
of Tsukuba.  
Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573 Japan

a) ijiro@cipher.risk.tsukuba.ac.jp

b) kanaoka@risk.tsukuba.ac.jp

c) okamoto@risk.tsukuba.ac.jp

d) kanayama@risk.tsukuba.ac.jp

ルを搭載している。生体認証は一般に生体情報取得に専用のデバイスを必要とするが、タッチパネルをデバイスとして用いることで、システム導入時のためのコストを抑えることが可能となる。

またタッチパネルによる操作の多くは公共の場においてもごく自然に行われており、認証システムの高い受容性が期待できるといえる。

#### 1.4 研究目的

タッチパネルによる手指の行動的特徴を用いた認証は近年提案された研究分野であり、特徴抽出や非類似度計算の手法、認証精度の向上など、実用化には未だ課題が残る。本稿ではタッチパネルでの手指の特徴抽出方法と認証精度の関係について、特に計測点の間引きによる認証精度変化を観測し、認証時間の削減を目指す。被験者実験を行った結果、認証精度を保ちつつ計測点を50%間引き、認証の高速化を図ることが可能であることを示した。

### 2. 関連研究

タッチパネルによる手指の行動的特徴を用いた関連研究として Sae-Bae 等の研究 [4], [5], および井芹等の研究 [6], [7] が挙げられる。いずれの研究においても被験者実験には Apple 社の iPad が用いられている。

#### 2.1 Sae-Bae 等の研究

Sae-Bae 等は 5 本の指の動きの個人差に着目し、それらの動きの特徴に応じた分類を行うとともに、22 種類のジェスチャを定義した。特徴抽出にはタッチパネルで取得した座標・時刻の軌跡を用い、オンライン署名に用いられる手法 [8] を用いて 2 つのジェスチャ間の距離を定義し、登録データと検証時の入力データとの非類似度計算を行っている\*1。

各々のジェスチャに対する被験者実験を行った結果、平均して EER が約 10% となることを示し、またジェスチャ 2 つの組合せで EER が 5% となることを示した。認証精度評価に用いる EER については 3.2 節に記す。

#### 2.2 井芹等の研究

井芹等は Sae-Bae 等と同時期に、独立して手指の行動的特徴を用いた認証手法を提案した。Sae-Bae 等と同様に 5 本の指の動きの個人差に着目した生体認証であるが、彼らはジェスチャとして開いた手をすばめるような動きを対象を絞って認証を行なっている。特徴抽出にはタッチパネルで取得した座標のうち各指が最初に触れた位置（始点）、および最後に触れた位置（終点）を利用する。ジェスチャ

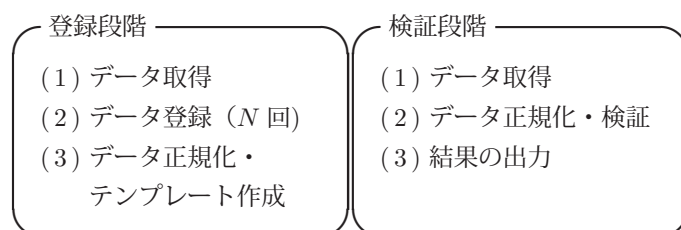
を行った時間と併せ、登録データと検証時の入力データとの非類似度計算を行なう。彼らは始点に 2 倍の重み付けを行うことで認証精度向上を図り、被験者実験を行った結果 EER は最低で 2.28% となることを示している。EER については 3.2 節に記す。

### 3. Sae-Bae 等の手法に基づく検証システム

被験者実験に用いた生体認証の検証システムの概要と要素技術を述べる。検証システムは Sae-Bae 等の手法に基づいて構成した [4], [5]。しかし Sae-Bae 等の手法には技術的詳細に関する記述が不足しているため、不明確な部分は推定をすることでシステムの実装を行なっている。

#### 3.1 システム概要

検証システムは次のように登録段階、認証段階の 2 つのステップで構成される。



##### 3.1.1 データ取得・登録

ユーザーがタッチパネル上で行ったジェスチャから、5 本の指の  $x, y$  座標の軌跡、タイムスタンプ（各タッチの時刻情報）をすべて取得する。この時、各指の対応付けを行うラベルを以下のように付与する。

##### ラベル付け手順

- (1) 5 点の中心点を求める\*2
- (2) 中心点とある基準位置との角度から各点を円順に並び替える
- (3) 各々の点に対し最近傍点を求め、最も最近傍点の遠い点を親指と推定する
- (4) 親指を先頭に並び替える

生体情報は完全に再現することができないため、データの登録は  $N$  回行う。

##### 3.1.2 データ正規化

取得した軌跡の正規化を行う。以下ラベル付けされた各指  $i (= 1, 2, 3, 4, 5)$  における  $j (= 1, 2, \dots)$  番目の計測点を  $(x(i, j), y(i, j))$  と表記する。

まず、親指が最初に触れた座標を原点とするよう位置の正規化を行う。正規化後の座標を  $(x'(i, j), y'(i, j))$  とすると以下の式のように表される。

\*2  $j$  番目の計測点の中心点  $(x_{jc}, y_{jc})$  は次のように求める。

$$(x_{jc}, y_{jc}) = \left( \frac{1}{5} \sum_{i=1}^5 x(i, j), \frac{1}{5} \sum_{i=1}^5 y(i, j) \right)$$

\*1 オンライン署名は、タブレットなどの端末に筆記された署名を利用し個人の同定を行う認証手法である。パラメータには座標や筆圧、筆記時間、筆記速度などの複数の情報を用いることができる。

$$x'(i, j) = x(i, j) - x(1, 1)$$

$$y'(i, j) = y(i, j) - y(1, 1)$$

位置に関する正規化後の最初に人差し指が触れた座標と、原点を結ぶ直線との角度が0度となるよう回転に関する正規化を行う。正規化後の座標を  $(x''(i, j), y''(i, j))$  とすると以下の式のように表される。

$$\theta_I = \tan^{-1} \frac{y'(2, 1)}{x'(2, 1)}$$

$$(R(i, j), \theta(i, j)) \xleftarrow{\text{Polar}} (x'(i, j), y'(i, j))^{*3}$$

$$(x'''(i, j), y'''(i, j)) \xleftarrow{\text{Cartesian}} (R, \theta - \theta_I)^{*4}$$

各指の軌跡の  $x, y$  座標の取りうる値が  $(0 \sim 1)$  の範囲となるよう経路の正規化を行う。正規化後の座標を  $(x'''(i, j), y'''(i, j))$  とすると以下の式のように表される。

$$x'''(i, j) = \frac{x''(i, j) - \min_j(x''(i, j))}{\max_j(x''(i, j)) - \min_j(x''(i, j))}$$

$$y'''(i, j) = \frac{y''(i, j) - \min_j(y''(i, j))}{\max_j(y''(i, j)) - \min_j(y''(i, j))}$$

### 3.1.3 テンプレート作成・検証

正規化を行った軌跡に対し2つのジェスチャデータ間の距離を定義し、テンプレート作成、および検証を行う。2つのジェスチャデータ間の距離は動的時間伸縮法 (DTW<sup>\*5</sup>) によって定義する。

DTW とは計測点数の異なるペア間の相違度計算法であり、一方の1点をもう一方の複数の点に対応づけることで時間的な伸縮を許容できる [9], [10]。DTW に基づく2ジェスチャ間の距離を次のように求める。

時系列データ  $A := \{a_1, a_2, \dots, a_M\}$ ,  $B := \{b_1, b_2, \dots, b_N\}$  間の DTW に基づく距離  $D(A, B)$  を求めることを考える。A, B の対応付けをワーピングパス  $F$  とよび、 $M \times N$  平面上の格子点  $f_k = (m_k, n_k)$  の時系列で表す。

$$F = f_1, f_2, \dots, f_K$$

ワーピングパスの例を図1に示す。

$a_{n_k}, b_{m_k}$  間の距離を  $d(f_k) = |a_{n_k} - b_{m_k}|$  とすると、 $F$  の評価関数  $\Delta(F)$  は次式で表される。

$$\Delta(F) = \sum_{k=1}^K d(f_k) \omega_k$$

この値が小さいほど A, B が類似していることを表す。極端な伸縮を防ぐために整合窓 ( $|m_k - n_k| \leq r$ ) の条件下で  $\Delta(F)$  を最小化する。ただし  $\omega_k$  は  $f_k$  に関する正の重みで

\*3 変換 “ $\xleftarrow{\text{Polar}}$ ” は極座標への変換を表す。

\*4 変換 “ $\xleftarrow{\text{Cartesian}}$ ” はデカルト座標への変換を表す。

\*5 Dynamic Time Warping

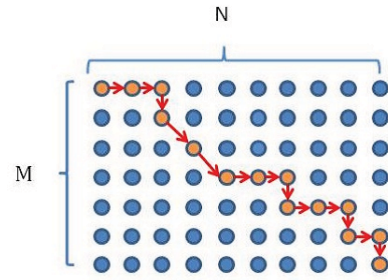


図1 DTWにおけるワーピングパスの例  
Fig. 1 An Example of Warping Path of DTW

以下の式を満たす。

$$D(A, B) = \min(\Delta(F))$$

$$\begin{cases} \omega_k = (m_k - m_{k-1}) + (n_k - n_{k-1}) \\ n_0 = m_0 = 0 \end{cases}$$

この問題には通常、時間計算量が  $O(MN)$  である動的計画法が用いられる。

本稿における時系列データはジェスチャを行った座標軌跡  $G$  である。 $G$  は次のように定義する。

$$G := \{Touch_1, Touch_2, \dots, Touch_j, \dots\}$$

$$Touch_j := \{(x_1, y_1), \dots, (x_5, y_5)\}$$

$$(x_i, y_i) = (x(i, j), y(i, j))$$

DTW に基づく距離を計算する際は直前の座標との差分  $\Delta Touch$  を利用する。差分に置き換えたものを  $G'$  として新たに次のように定義する。

$$G' := \{\Delta Touch_1, \Delta Touch_2, \dots, \Delta Touch_j, \dots\}$$

$$\Delta Touch_j := \{(\Delta x_1, \Delta y_1), \dots, (\Delta x_5, \Delta y_5)\}$$

$$(\Delta x_i, \Delta y_i) = (x(i, j) - x(i, j-1), y(i, j) - y(i, j-1))$$

ただし  $\Delta Touch_1 := \{(0, 0), \dots, (0, 0)\}$  とする。

2つのジェスチャ  $G'_A, G'_B$  におけるワーピングパスの格子点  $f_k = (\Delta Touch_{A_{m_k}}, \Delta Touch_{B_{n_k}})$  に対し、 $\Delta Touch_{A_{m_k}}, \Delta Touch_{B_{n_k}}$  間の距離  $d(f_k)$  を次のように定義する。この時  $(1 \leq m_k \leq M)$ ,  $(1 \leq n_k \leq N)$  であり、 $G_A$  に関するマルチタッチ座標の差分を  $\Delta Touch_A$ ,  $x, y$  座標を  $x_A, y_A$  等と表す。

$$d(f_k) = |\Delta Touch_{A_{m_k}} - \Delta Touch_{B_{n_k}}|$$

$$= \sqrt{\sum_{i=1}^5 ((\Delta x_{A_{m_k}} - \Delta x_{B_{n_k}})^2 + (\Delta y_{A_{m_k}} - \Delta y_{B_{n_k}})^2)}$$

$$(\Delta x_{A_{m_k}}, \Delta y_{A_{m_k}}) = (x_A(i, m_k) - x_A(i, m_k - 1), y_A(i, m_k) - y_A(i, m_k - 1))$$

$$(\Delta x_{B_{n_k}}, \Delta y_{B_{n_k}}) = (x_B(i, n_k) - x_B(i, n_k - 1), y_B(i, n_k) - y_B(i, n_k - 1))$$

テンプレート作成時は  $N$  個の登録データ  $G := \{G_1, G_2, \dots, G_N\}$  および 2 ジェスチャ間距離  $D()$  を用いて以下の式を計算する。

$$\overline{G_{\min}} = \sum_j \min_i (D(G'_i, G'_j)) / N$$

$$\overline{G_{\max}} = \sum_j \max_i (D(G'_i, G'_j)) / N$$

$$\overline{G_{ref}} = \sum_j D(G'_{ref}, G'_j) / (N - 1)$$

上式における  $ref (1 \leq ref \leq N)$  は  $N$  個のデータ中で、自身以外とのデータ間距離  $D()$  の平均値が最も小さいものの番号を表している。

テンプレートデータは登録時に取得した  $N$  個のジェスチャデータに上式の結果を加え、以下のように与える。このように登録データ同士の距離を計算し検証時に用いることで、ユーザーごとに異なる最適なしきい値の正規化を行い、しきい値をユーザーによらず定めることが可能となる [4], [8]。

$$Template := \{G_1, G_2, \dots, G_N, \overline{G_{\min}}, \overline{G_{\max}}, ref, \overline{G_{ref}}\}$$

検証時は検証時に入力されたジェスチャ  $G_{ver}$  とテンプレートとの非類似度 ( $score()$ ) を以下のように計算する。

$$score(Template, G_{ver}) = \frac{\min_i (D(G'_i, G'_{ver}))}{\overline{G_{\min}}} + \frac{\max_i (D(G'_i, G'_{ver}))}{\overline{G_{\max}}} + \frac{D(G'_{ref}, G'_{ver})}{\overline{G_{ref}}}$$

### 3.1.4 結果の出力

検証時に計算結果  $score()$  が、特定のしきい値を下回れば入力された結果は容認され、上回れば拒否される。しきい値の決定は認証精度評価に用いられる他人受入率 (FAR\*<sup>6</sup>)、および本人拒否率 (FRR\*<sup>7</sup>) に基きシステムの要件に応じて決定する [11]。

## 3.2 精度評価方法

本稿における生体認証の検証システム評価は等誤り率 (EER\*<sup>8</sup>) を用いて行う。これは、あるしきい値に対する他人受入率 (FAR)、および本人拒否率 (FRR) が等しくなる時の割合である。この割合が小さいほど良い精度であるとみなすことができる。

\*<sup>6</sup> False Acceptance Rate

\*<sup>7</sup> False Rejection Rate

\*<sup>8</sup> Equal Error Rate

### 3.2.1 他人受入率 (FAR)

FAR は誤って他人を受け入れてしまう割合を表す。しきい値  $x$  に対する要素の個数を  $\#\{\}(x)$ 、本人以外の認証データを偽のケースとすると次のように求めることができる。

$$FAR(x) = \frac{\#\{\text{検証システムが容認した偽のケース}\}(x)}{\#\{\text{偽のケース}\}(x)}$$

### 3.2.2 本人拒否率 (FRR)

FRR は誤って本人を拒否してしまう割合を表す。FAR と同様に、本人の認証データを真のケースとすると次のように求めることができる。

$$FRR(x) = \frac{\#\{\text{検証システムが否認した真のケース}\}(x)}{\#\{\text{真のケース}\}(x)}$$

## 4. 計測点間引き

本稿における間引きとは、得られた計測点の軌跡を一定の間隔で取り除くことを指す。間引き率  $x\%$  ( $0 \leq x \leq 100$ ) は、計測したすべての点から  $x\%$  の点を取り除くことを表している。計測点間引きアルゴリズムの概要は以下のとおりである。

### 計測点間引きアルゴリズム

- (1) 間引き率  $x$  から  $K$  を求める。  $K$  は  $x \geq 50$  のとき間引き後の計測点数とし、  $x < 50$  のとき間引きを行う計測点数とする (端数は四捨五入)\*<sup>9</sup>
- (2) 計測点全体を  $K$  の群に分割する (第  $K$  群については他の群と異なる要素数を認める)
- (3) 事前に始点・終点の座標を選択する
- (4) 各群内の要素数の半分の点 (端数は四捨五入) を選択する
- (5)  $x \geq 50$  のとき
  - 第  $k$  群 ( $1 \leq k \leq K$ ) における “選択されていない点” を間引く (第 1 群,  $K$  群についてはそれぞれ始点・終点がすでに選択されている)
  - すべての  $k$  に対し同様の処理を行う
- (6)  $x < 50$  のとき
  - 第  $k$  群 ( $1 \leq k \leq K$ ) における “選択されている点” を間引く (第 1 群,  $K$  群を含む)
  - すべての  $k$  に対し同様の処理を行う
  - 例外として、始点・終点を選択されていた場合、この間引き処理は行わない

## 5. 実験

間引きによる認証の高速化を評価するため被験者実験を行った。以下実験環境および実験の概要について記す。

\*<sup>9</sup> 間引き前の計測点数を  $J$  とすると  $K$  は以下のように与えられる。

$$K = \begin{cases} J(100 - x)/100 & (x \geq 50) \\ Jx/100 & (x < 50) \end{cases}$$



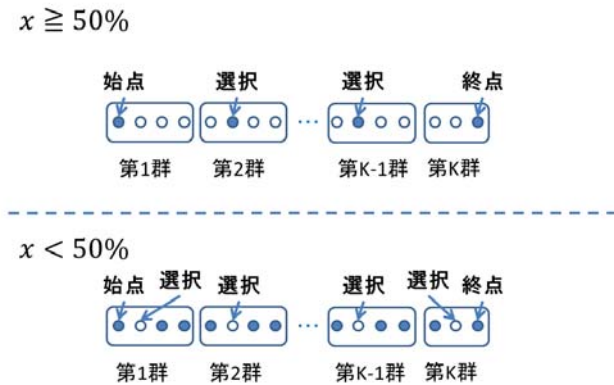


図 2 計測点間引きアルゴリズムの概要  
Fig. 2 Summary of Cull Algorithm

表 1 実験機器の仕様  
Table 1 A Spec of Device

ハードウェア	iPad
モデル	MB294J
OS	iOS 5.1
CPU	1.0GHz Apple A4 カスタムデザイン
メモリ	256MB
ディスプレイ	9.7inch
解像度	1024 × 768
マルチタッチ	一度に 5 点の追跡可能
計測点取得頻度	20-65[point/sec]

表 2 開発環境  
Table 2 Development Environment

計算機	MacBook Pro (15-inch Core 2 Duo)
OS	Mac OS X 10.7.4
CPU	2.33GHz Intel Core 2 Duo
メモリ	2GB 667MHz DDR2 SDRAM
開発環境	Xcode 4.3.2 (4E2002)
言語	Objective-C

### 5.1 実験環境

本稿では被験者実験時に Apple 社の iPad を用いてタッチパネルから指の行動的特徴を取得した。実験に用いた機器の詳細を表 1 に、また開発環境および計算機に関する情報を表 2 にそれぞれ記す。

### 5.2 ジェスチャの描画と遅延

本実験では被験者にジェスチャを視認させるためタッチされた軌跡の描画を行なっている。しかし描画を行うことで計測間隔の遅延が発生する。

本システムの環境ではデバイスで検知したタッチ情報は iOS の API 関数である touchesMoved メソッドによって  $x, y$  座標、タイムスタンプとして取得している。このメソッドはタッチパネル上に指が触れたまま移動したことを

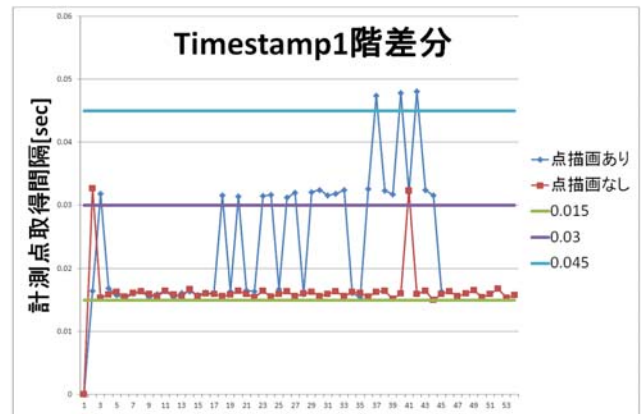


図 3 計測点間隔の比較  
Fig. 3 Comparison of Measurement Point Interval

検知した場合呼び出される。タッチパネル上への計測点の描画は touchesMoved メソッドによって新たな座標を取得する度に画面の更新を行い、同じく iOS の API 関数である drawRect メソッド内の記述にしたがって再描画される。したがって計測点数の増加に伴い再描画にかかる時間は増加する。計測点間隔に遅延が発生する原因は再描画処理中にタッチの検出をおこなっていないことが原因であると推測される。

Apple 社の仕様書 [12] には touchesMoved メソッドの呼び出し頻度に関する記載はなかったため、描画の有無によるタイムスタンプの 1 階差分によって呼び出し頻度とその遅延について調べた。図 3 における点描画なしのグラフを見ると touchesMoved メソッドの呼び出し頻度はおおよそ 0.015sec/回であり、描画を行うとほぼ定数倍のステップ状に遅延が起こることがわかった。

### 5.3 Timestamp を用いた $\Delta Touch$ の補正

計測点の取得遅延が発生すると DTW に基づく距離計算を行う際の  $\Delta Touch$  の値に影響をあたえるため、DTW を行う前に差分値の補正を行う。計測点差分  $\Delta(x_i, y_i)$  におけるタイムスタンプの 1 階差分を  $\Delta t_j (= t_j - t_{j-1}, \Delta t_1 = 0)$  とおき、計測点差分の補正を次のように行う。

$$\Delta(x_i, y_i) = \Delta(x_i, y_i) \times \left( \frac{\min_j(\Delta t_j)}{\Delta t_j} \right)$$

ただし  $\min_j(\Delta t_j) > 0$  とする。

### 5.4 実験概要

実験は、被験者に認証システムを実装したデバイスを手渡し、被験者にとって行きやすい自然な姿勢、動きで認証動作を行なってもらった。被験者には実験の目的や概要を伝え、各ジェスチャに対し以下の手順で実験を行った。

実験手順

- (1) 実験で行なってもらうジェスチャについての説明
- (2) 数回の練習
- (3) データの登録 (10回)

## 6. 実験を行ったジェスチャ

実験は Sae-Bae 等の定義するジェスチャから 3つを選択し、新たに 9 のジェスチャを加え、計 12 のジェスチャで実施した。

### 6.1 Sae-Bae 等の定義するジェスチャ

Sae-Bae 等の定義するジェスチャから、以下の条件を考慮して CCW, Close, Drag のジェスチャを選択した。

- 低い EER である
- 選択したジェスチャが、Sae-Bae 等の行ったジェスチャの分類に対し、お互い異なる分類に属する
- Sae-Bae 等の実験で実施した被験者人数が多い

実験を行った Sae-Bae 等のジェスチャ

- CCW** すべての指を反時計回りに回転させる
- Close** すべての指を掌の中央へ向けて動かす
- Drag** すべての指を上から下へ並行に移動

### 6.2 新たに追加したジェスチャ

Sae-Bae 等の定義するジェスチャは一樣な動きで行われるものが多いため、本研究では新たに 9 つのジェスチャを定義した。ジェスチャはなめらかな曲線や角を持つ図、円や四角形など特定のイメージを持つことのできるものと特定名称のない図とを用意した。

また、ジェスチャは Sae-Bae 等との比較を行うため一筆書きで行うことを条件とした。新たに定義したジェスチャ及びジェスチャ名称を図 4 に示す。また図 4 にないジェスチャ “Original” は、ユーザーが任意に定めたジェスチャを表す。

## 7. 評価・考察

### 7.1 実験結果

間引き率を 0 から 87.5% まで 12.5% ずつ変化させた時の EER 変化、及び認証時間変化の様子をそれぞれ図 5, 6 に示す。

この結果から間引き率の増加に応じて認証時間は線形に短くなることにに対し、EER は線形な変化をしていないことが分かる。間引き率に対する認証時間-EER の関係を図 7 に示す。また同じ間引き率であったときの各ジェスチャ認証時間、EER の平均値によるグラフを図 8 に示す。

図 8 のグラフの形状から一定の間引き率までは EER 変

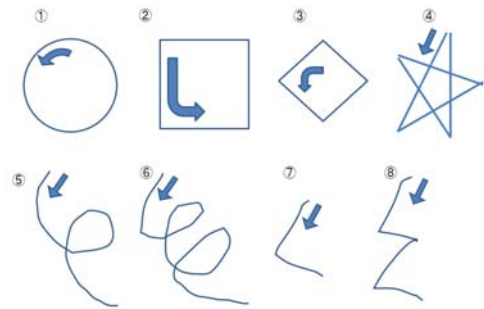


図 4 定義したジェスチャ  
Fig. 4 Gesture Definition

ジェスチャ名

- |             |                             |
|-------------|-----------------------------|
| (1) Circle  | (5) 1Loop                   |
| (2) Square  | (6) 2Loop                   |
| (3) Diamond | (7) 1Ell                    |
| (4) Star    | (8) 2Ell                    |
|             | (9) Original (ユーザー任意のジェスチャ) |

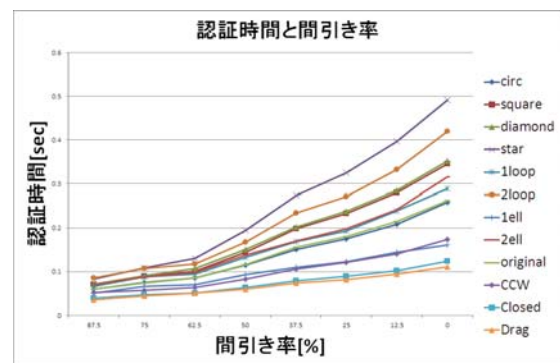


図 5 間引き率変化による認証時間の変化  
Fig. 5 Behaviour of Cull Rate and Authentication Time

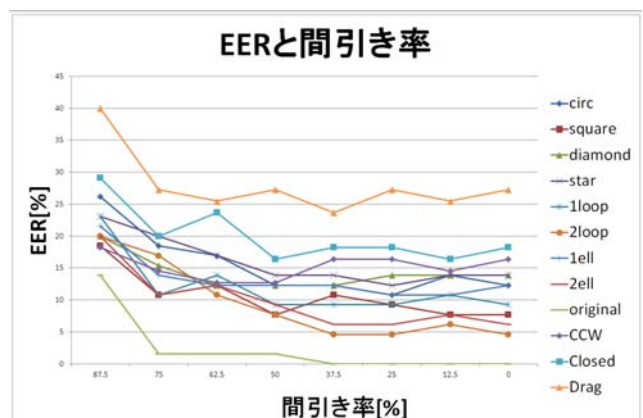


図 6 間引き率変化による EER の変化  
Fig. 6 Behaviour of Cull Rate and EER

動は小さく、それ以上では EER が増加し認証精度が下がる傾向が見られることが分かる。

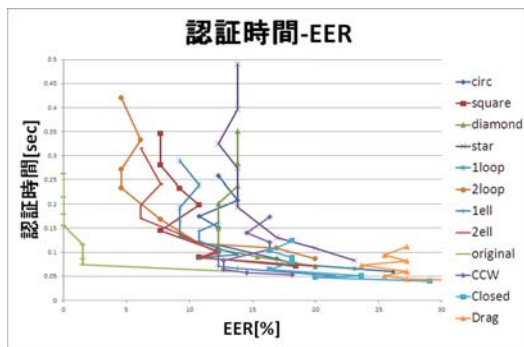


図 7 認証時間-EER 関係図  
 Fig. 7 Behaviour of Authentication Time and EER

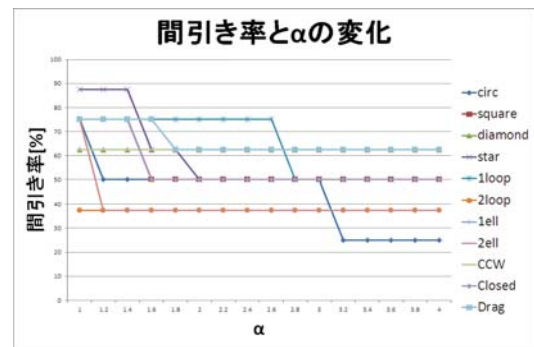


図 9  $k$  を最小化する間引き率と  $\alpha$  の関係  
 Fig. 9 Relation Between Cull Rate and  $\alpha$  That Minimizes  $k$

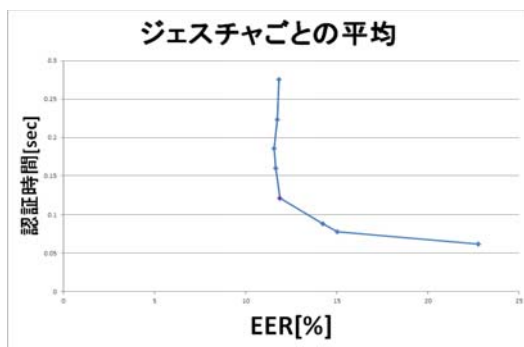


図 8 同間引き率におけるジェスチャごとの平均値  
 Fig. 8 Average for Each Gesture of Same Cull Rate

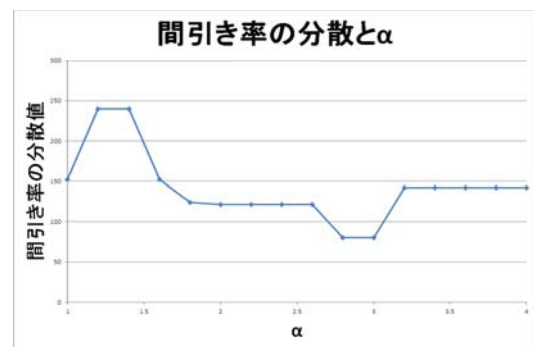


図 10  $\alpha$  による最適な間引き率の分散値  
 Fig. 10 Variance of Optimal Cull Rate Due to  $\alpha$

## 7.2 間引き率評価式 $k(\alpha)$

システムに対し、最適な間引き率を与えるための評価式を考える。EER と認証時間はトレード・オフの関係であり、安全性の観点においては EER が低いことが望ましく、システムの観点においては認証時間が短いことが望ましい。そこで EER・認証時間を評価するための評価指標  $k(\alpha)$  を次のように定義する。

$$k(\alpha) = (\text{EER}[\%])^\alpha \times (\text{認証時間})[\text{sec}]$$

上式における  $k$  を最小化するような間引き率を選択することが望ましい。ここで  $\alpha$  はシステムの要件によって定まる値である。 $\alpha$  を大きく取るとより EER の結果を重視するような結果となる。

## 7.3 評価指標 $k(\alpha)$ の $\alpha$ 値における最適な間引き率と認証の効率評価

各々のジェスチャの  $k$  を最小化する間引き率の  $\alpha$  による変化の様子を図 9 に、各々のジェスチャの  $k$  を最小化する間引き率の分散を図 10 に示す。ジェスチャ “Original” については各ユーザごとに異なる特異なジェスチャであるため  $k(\alpha)$  による評価から除外する。

図 10 より、ジェスチャごとの最適間引き率の分散に小さい、評価指標  $k(\alpha = 3)$  の変化の様子を図 11 に示す。最小値近傍の図 12 から、EER の安定しない Drag のジェス

チャを考慮から除外すると  $k(\alpha)$  は間引き率は 50% のときおおよそ最小値となる。本システムにおいて間引き率を 50% とした時、図 7 より、ジェスチャ毎に平均して認証時間は 44.2% の削減が望める事に対し、EER は最大 0.31% 程度の変化にとどまり、安全性を維持しつつ認証時間を大きく削減することが可能である。

## 8. まとめ

本稿はタッチパネルを用いた手指の行動的特徴に基づく生体認証において、タッチパネルから取得する計測点の間引きを行うことで精度を維持しつつ認証の高速化を図ることを目的とし、先行研究である Sae-Bae 等の方式に基づいて検証システムの実装を行った。また、一様な動きである Sae-Bae 等のジェスチャの定義に対し様々な特徴を持つジェスチャを新たに定義し、Sae-Bae 等のジェスチャと併せ比較を行った。

システムに対し最適な間引き率を与えるための評価式  $k(\alpha)$  を提案し、本システムにおいては間引き率 50% が適切であるという結果を与えた。このとき、EER 変動が 0.31% 程度の変化にとどまり、また認証時間は 44.2% の削減を見込める結果を示すことができた。

今後の課題としてなりすまじや、時間変化による認証精度評価、特徴抽出において個人の識別への影響の強い指標の調査、また、圧力といった、より多くのセンサを用いて

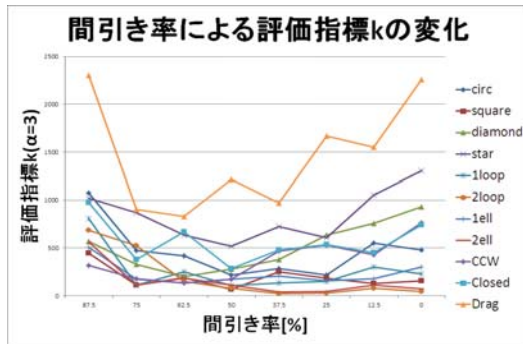


図 11 間引き率による評価指標  $k(\alpha = 3)$  の変化  
 Fig. 11 Behaviour of  $k(\alpha = 3)$  Due to Cull Rate

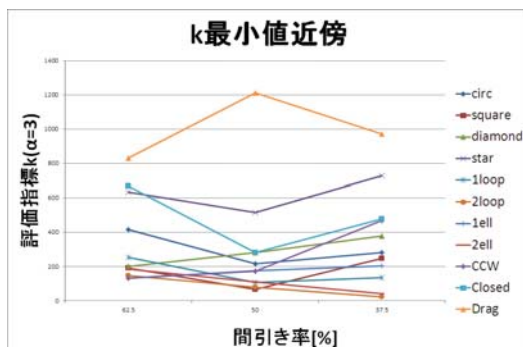


図 12 図 11 の最小値近傍図  
 Fig. 12 Neighbourhood of Minimum  $k$  in Figure 11

精密な生体情報を取得することによる認証精度変化の調査などがあげられる。

### 参考文献

[1] Google : OUR MOBILE PLANET  
 入手先 〈<http://www.ourmobileplanet.com/ja/>〉  
 (2013.02.04)

[2] スマートフォンネットワークセキュリティ実装ガイド  
 入手先 〈<http://www.jssec.org/dl/NetworkSecurityGuideB1.pdf>〉  
 (2013.02.17)

[3] 内田薫：携帯機器におけるバイオメトリクス認証の利用、  
 バイオメトリクスセキュリティの実用化最前線小特集  
 (2007).

[4] Napa S.B. , Kowsar A., Katherine I., and, Nasir M. :  
*Biometric-Rich Gestures: A Novel Approach to Authentication on Multi-touch Devices*, CHI 2012, May 5-10, Austin, Texas, USA (2012).

[5] Napa S.B. : *Biometric-Rich Gestures-A novel approach to authentication on multi-touch devices-*, Kaspersky Lab, "IT Security for the Next Generation", American Cup, New York, 9-11 November (2011).  
 入手先 〈<http://www.kaspersky.com/images/Napa%20Sae-Bae-10-108178.pdf>〉 (2013.02.17)

[6] 井芹隼人, 岡本栄司：タッチパネルを用いた行動的特徴に基づくバイオメトリクスに関する一考察, Computer Security Symposium 2011, pp.84-88, October 2011.

[7] 井芹隼人：タッチパネルを用いた行動的特徴に基づくバイオメトリクスに関する研究, 筑波大学大学院 博士課程システム情報工学研究科 修士論文 (2012).

[8] Alisher K. and Berrin Y. : *Biometric Authentication Using Online Signature*, ISCIS, Springer LNCS-3280

373380, Springer-Verlag (2004) , pp.373-380 (2004).

[9] HIROAKI S. and SEIBI C. : *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*, IEEE, Transactions on Acoustics, Speech and Signal Processing, 26 (1) . , pp.43-49 (1978).

[10] 山田 悠, 鈴木 英之進, 横井 英人, 高林 克日己：動的  
 時間伸縮法に基づく時系列データからの決定木学習, 情  
 報処理学会研究報告. ICS, [知能と複雑系] 2003 (30) ,  
 pp.141-146 (2003).

[11] 独立行政法人 情報処理推進機構：生体認証導入・運用の  
 ためのガイドライン (2007).

[12] iOS View プログラミングガイド  
 入手先 〈[https://developer.apple.com/jp/devcenter/ios/library/documentation/ViewPG\\_iPhoneOS.pdf](https://developer.apple.com/jp/devcenter/ios/library/documentation/ViewPG_iPhoneOS.pdf)〉  
 (2013.02.17)