

自動的に特徴点を生成する 生体認証システム

稲葉勇希[†] 堂 蘭 浩[†]

近年スマートフォン等の情報端末の普及により個人情報には常に危険にさらされている状態である。本研究は、スマートフォン等の携帯端末において、使いやすく性能の高い認証システムを実現することを目標としている。その第一段階として、利用者が自由にタッチパネルにかいた図形から特徴点を生成し、その図形の再現の手助けを行い、同時に生体認証を行うシステムの開発を行っている。

A Biometric System which Generate Feature Points Automatically

Yuki Inaba[†] and Hiroshi Dozono[†]

Recently, the personal information is threatened with the popularisation of the mobile devices like smart phones. The purpose of this research is to develop the authentication system which has both features of high accuracy and user friendly. As the first step of this research, a biometric authentication system, which generates feature points automatically from the pattern drawn on the touch panel to help the user for re-drawing the pattern and to execute biometric authentication, is under development.

1. はじめに

近年社会の情報化が急激に進み電子情報の利用率は高まる一方である。特に個人情報に関しては、団体や会社という個人のグループに留まらず国が国民の情報を電子化するなど行政のツールとして利用するまでに至っている。

しかしその利用率が直接、社会的に電子情報の取り扱いの知識が高くなった事を示す訳ではない。その一例として挙げられるのがスマートフォンやタブレット端末などである。

小さなパソコンとも呼ばれるスマートフォンは近年爆発的に普及しており、人口に対する所持率に至ってはここ数年で数倍にまで伸びた。普及の理由は今までの携帯端末では考えられないほどの情報処理能力と 3G 回線などネットワーク、またインターフェース等の使いやすさもその要因となっている。しかし、その情報処理能力やネットワークは同時に端末が所有する個人情報量を莫大な物とし、使いやすさは情報に関する知識がないユーザへの普及を促してしまい、結果として情報の取り扱いに対して無防備な利用者を爆発的に増加させている。

基本的にこういう被害から身を守るには自分で知識を身に付け、常に気を置き対処をし続けるしかないのだが、子供、老人、女性など、このような機器を用いる事が苦手である人等は知識を身に付ける事が困難である。そこで本研究では利用者が情報の操作や盗難等の危険性に意識せずに、システムを利用する事ができるシステムが重要であると考え、タッチパネルを搭載した携帯端末を対象に知識が無くても利用し易い個人情報端末向けの認証システムの開発を行うものとする。

タッチパネルを用いた認証方式の代表的なものとしては、アンドロイド端末 [1] で用いられている点を自分の設定した順番で結ぶものがある。ただし、この方法で作成可能なパターンはあまり多くはなく、単純なものでは他人に見られるとすぐにわかってしまう可能性もあり、逆に複雑なものは、点が等間隔にならんでいるのみであり、本人もパターンが覚えにくいという問題もある。これに対し、決められた図形をなぞる際の筆速と筆圧を用いて生体認証を行う方法が提案されている[2]が、最近のスマートフォンなどで筆圧が検出可能なパネルを搭載したものは見当たらない。今回はアンドロイドの認証方式に筆速を組み合わせて、認証の強度を強化した方式を提案し、また、利用者の自由度を高めるために、自由に書いた図形から自動的に特徴点を生成して認証を行う方法を提案する。

[†]佐賀大学大学院工学系研究科先端融合工学専攻
Department of Advanced Technology Fusion, Graduate School of Science and Engineering Saga University

2. 筆速を用いた生体認証アプリケーション

タッチパネルを用いた認証を行う上で、今回作成したシステムには筆速を用いた生体認証を用いる事にした。

2.1 生体認証とは

まず詳細な認証システムの説明に入る前に生体認証について説明を行う。

生体認証[3]とは大きく言うと従来の様にパスワードや個人番号などのデジタルなデータを用いるのではなく、個人の体の部分的な特徴や行動を行う際の癖等を判断材料として個人の判別を行うある意味アナログな認証システムである。認証には主に指紋、光彩、筆跡などが特徴量として用いられる。

セキュリティに生体認証を用いる利点として挙げられるのは、どの特徴にしても情報量が多すぎて『認証情報複製が困難』だということ。また体に組み込まれた情報であるためパスワード等の解除キーと違い『認証情報そのものの盗難が出来ない』という点である。つまり生体認証を組み込んだセキュリティでは他人が本人に成りすます事が非常に困難であるという特徴をもつ。

また、生体認証は身体的特徴量を用いた認証方式と行動的特徴量を用いた認証方式に大別される。身体的特徴量は指紋、光彩、顔、静脈パターン等の個人を判別するのに可能な身体の特徴を用いて認証を行う方法である。一般的に認証の精度は高いが、センサとして特別なハードウェアが必要となるものが多く、写真や模型によりなりすましが行われる可能性もある。行動的特徴量は、筆跡、歩行、音声などの個人の行動に基づく特徴を用いて認証を行う方法である。一般的に認証の精度は身体的特徴量に比較すると低い、スマートフォンなどに装備されたセンサ類で測定可能なものが多く、行動に基づくものであるため、なりすましもしにくい。本研究では行動的特徴量のひとつとして、筆速を用いた認証方式を用いる。

2.2 筆速による生体認証

今回生体認証として筆速の特徴量を用いた理由としては端末の互換性が挙げられる。

前述したとおり本研究の生体認証システムは手軽に使える事を大前提として携帯端末に搭載する物を考案している。そうすると端末の携帯性を損なわない様にすることは特徴量の取得に専用のデバイスが必要とする指紋や光彩は使えず、必然的に携帯端末に搭載されたインターフェースから特徴量を取得できる物を選択しなければならない。そこで端末のタッチパネルから筆跡による特徴量を取得する事にした。しかしそれでも端末によって取得できる情報に差があり、筆圧などでは取得できない場合がある。よってタッチパネルならば間違いなく取得出来る情報であろう接触座標を用い

て筆速による生体認証を行う事にした。

今回、2点 a,b 間の筆速は以下の式で定義する。

$$\text{筆長} = \sum_{a \rightarrow b} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$
$$\text{筆速} = \text{筆長} / T$$

タッチパネル上に指で曲線を描写すると、システムは a,b 間の曲線を微小な間隔で取得した座標 (x_i, y_i) を記憶する。それらの点ごとに筆記距離から計算した筆長を筆記時間 T で割ることで a,b 間の筆速を取得する。

2.3 認証パスに与える閾値

生体認証を行う際にはあらかじめ認証の基準になる筆速の領域を作っておかなければならない。この領域とは実際に利用者から取得した筆速に閾値を与えて値に余裕を持たせたものである。この閾値をどれだけ与えるかによって生体認証の精度が大幅に変化する。閾値は各点間の筆速と認証パターン全体の平均の筆速に与えるものとする。

今回作成した生体認証システムには閾値を認証結果に応じて変化させる仕組みを搭載した。

認証成功時 更新された閾値 ← 閾値 - 40 / (1 + 認証失敗回数)

認証失敗時 更新された閾値 ← 閾値 + 50 / 認証失敗回数

予め閾値を大きく設定しておき認証を成功した場合は閾値を小さくし、認証に失敗した場合は一旦閾値を戻して細かい値でより適切な値に調整していくという仕組みである。

2.3.1 生体認証精度検証

実際に生体認証の精度を検証するため Android のロック解除システムを応用した生体認証アプリケーションを作成した。

アンドロイドの認証システムは、元々画面上の9つの点の結線パターンによるみ認証を行うシステムであるが、ここに筆速による生体認証を行うアルゴリズムを追加した。このアプリケーションを用いて決まったパターンによる認証を行い生体認証の精度を検証する。認証パターンには検証毎に各個人が設定したものをを用いる。認証の閾値はあらかじめ前述のアルゴリズムにより収束させておき、FAR、FRR は一人10回分のデータから用いて検証を行った。FAR は認証パターンが知られてしまった場

合を想定し、各個人の認証パターンにつき7人分で実験を行った。

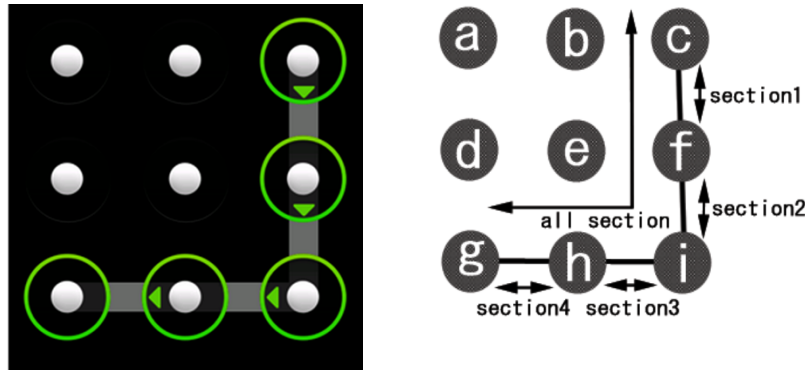


図1 認証画面と筆速の取得領域

表1 認証精度検証結果

	検証1	検証2	検証3	検証4
収束閾値	98	82	101	94
FRR[%]	22	0	22	11
FAR[%]	30	2	16	2

検証の結果検証2と検証4ではFAR,FRRの値がそれぞれほぼ10%以下という事で非常に優秀な値が出ている。しかし検証1と検証3においてはやや高い値となっている。ただ、認証パターンが知られた場合を想定し、筆速のみという単純な方式で認証を行っていることを考えると、妥当な結果といえる。また、検証中に利用者の数人からインターフェースがつかいづらい等の意見が出ていたため、特徴点を使いやすい配置にする事で結果が改善される可能性がある。特にFRRの値については、特徴点がいやすく配置されれば、ユーザの入力もしやすくなり、大幅に改善される可能性も考えられる。さらに、FRRの値が良くなれば閾値を小さく出来るため自然とFARも小さく設定可能であると考えられる。

3. 自動的に特徴点を取得するシステム

前節で述べたようにアンドロイドの認証システムに生体認証を組み合わせた場合、設定された点の配置が単純なため認証パターンのバリエーションが少ない、また、点が固定されているため、認証に使いやすい特徴点の配置が行いにくい、という問題があった。逆に、完全なフリーハンド曲線から特徴量を特定し、認証を行う事も難しいと考えられる。その理由は自由に曲線を描写出来てしまうと、利用者が毎回同じ筆跡を画面上に描写する事が困難になるからである。そこで、ある程度のアンカーとなる特徴点を設けて元の曲線に近いものをかいてもらう方法が考えられる。本研究では作成しているシステムでは、自動的にこの特徴点を抽出し、一筆書きの要領で結線する行動により、元の曲線に近い行動を再現する事が目標である。

3.1 特徴点の自動取得

特徴点の自動取得とは、画面上に描写されたフリーハンド曲線から特徴量の多い部分を探し出して、自動的にマーキングする事がである。これを軌跡再現の補助に用いれば、より利用者がストレスなく自分の描いた曲線を再現しやすくなり、軌跡の再現率が高まれば特徴量も安定し、自然と本人が認証を失敗する確率(FRR)を低下させることが出来ると考えている。

3.2 特徴点の取得方法

特徴点は筆速と同じく曲線上の座標から計算する。定間隔で3点を取得し、この3点から曲率を計算し、曲率の大きい部分をより特徴の大きい点としてマーキングを行う。なお取得する三点の間隔は3pixel毎で、描写が行われている限り継続して更新を行う。実際に特徴点を取得した様子を図2に示す。

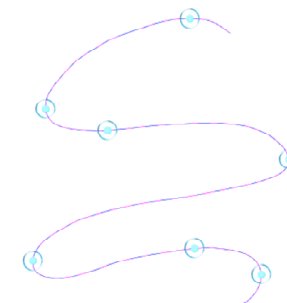


図2 特徴点取得例

3.3 問題点

3.3.1 手ブレの誤認識

この段階でまず明らかになった問題点として特徴点の誤認識がある。曲率の計算は

$$\text{傾き 1} = (Y_a - Y_b) / (X_a - X_b)$$

$$\text{傾き 2} = (Y_b - Y_c) / (X_b - X_c)$$

$$\text{傾き 1} - \text{傾き 2} = \text{曲率}$$

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以下のようにアルゴリズムを改良した。

$$\text{傾き 1} = (Y_a - Y_b) / (X_a - X_b)$$

$$\text{傾き 2} = (Y_b - Y_c) / (X_b - X_c)$$

$$\text{曲率 1} = \text{傾き 1} - \text{傾き 2}$$

$$\text{傾き 3} = (X_a - X_b) / (Y_a - Y_b)$$

$$\text{傾き 4} = (X_b - X_c) / (Y_b - Y_c)$$

$$\text{曲率 2} = \text{傾き 3} - \text{傾き 4}$$

曲率 1 および曲率 2 が 0.2 以上で特徴点としてマーキング

3.3.2 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特徴点とする事で対策を行った。統合点の座標は、元のマーキング点の座標と曲率から、以下の式によって計算する。

$$X \text{ 統合点} = X_a + (X_b - X_a) \times \{K_b / (K_a + K_b)\}$$

K_a : X_a の曲率

K_b : X_b の曲率

Y 統合点についても同様に計算する。

3.3.3 特徴点の見逃し

また、前述の二つの問題の様にマーキングの位置ではなく描写された曲線の中から特徴的な点を取り逃すというエラーが生じた。この問題が生じる原因として代表的な物は曲率の小さな曲線である。

このシステムは描写された曲線をそのまま見るわけではなく、取得した 3 点の座標から見ている。この三点の間隔が曲線の変化量に比べて狭すぎるとシステムは直線が続いていると判断してしまいマーキングをすることができない。そこで、ある程度の

期間マーキングが行われない場合、3pixel 毎の三点とは別に 13pixel で取得した三点からも特徴点の検出を行う様にした。これを行うとアルゴリズム自体は同じだが、曲線が緩くなった場合もある程度の曲率を検出する事が可能となり緩い曲線でも特徴点を検出する事が可能である。

実際にこれらの改善を行う前と行ったあとの特徴点の抽出の結果を図 3 に示す。左の改善前の図では明らかに特徴点が多すぎる部分があるが、右の図では適度に特徴点の数が抑えられており、また、曲率が低い部分での抽出も右図では行われており、アルゴリズムの修正が上手く働いていると考えられる。

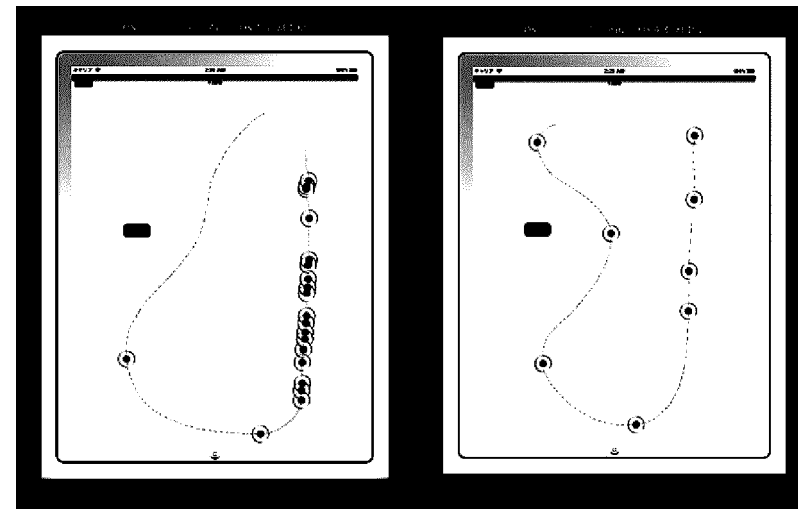


図 3 特徴点取得の結果：改善前(左)及び改善後(右)

図 4 に実際の特徴点取得の例を示す。

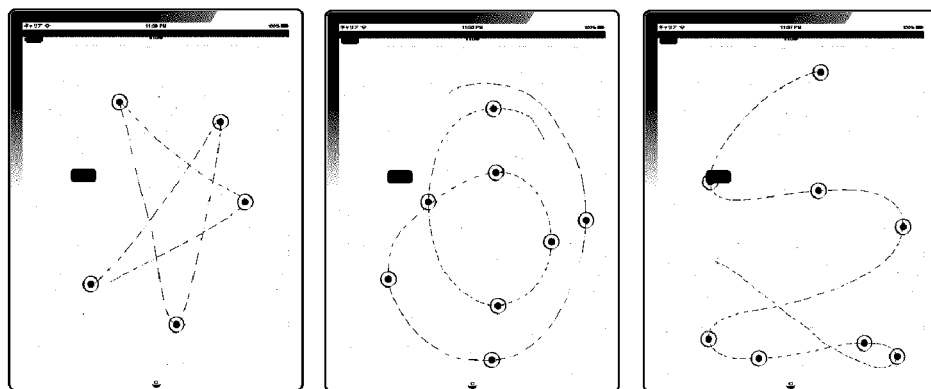


図4 特徴点取得結果の例

4. 認証システムの構成

これまでに説明してきた機能を用いて実際に認証システムを構成する。

4.1 認証手順

登録時

【step1】タッチパネル上に曲線を描写して特徴点をマーキングする。

【step2】取得した特徴点を用いて認証用のパスを作成する。認証パスとは特徴点の結線パターンと結線時の筆速を記録したもので、生体認証の基準となる筆速はこの段階で3回取得し、その平均値を認証に用いる。3回取得する間に結線パターンが違うものが含まれた場合エラーを返し【step1】からやり直す。

認証時

【step1】表示された特徴点を、登録時の【step2】通りに結線し認証を行う。

認証用パスは端末に保存しておき、それらを必要に応じて読み込む事で、1 端末でも複数人が利用できる様にしたいと考えている。

4.2 特徴点の役割

特徴点は前述した通り元の曲線を再現するためのアンカーなのだが、セキュリティの観点からはもう一つの役割を持つ。実際の認証画面では図5のように曲線は表示されず、特徴点しか表示されないようにしている。認証を成功させる為には【step2】で入力した順番で結線をする必要があるため、当然認証パスを作成した本人以外は結線順番が判らず認証を行う事が出来ない。この際、問題となるのが、特徴点から本人が元の曲線を再現できるか、また、特徴点のみから他人が曲線を再現できるかである。前者については認証システムを毎日使う等、使用の頻度が高ければ問題ないと考えられるが、後者については検証が必要であると考えられる。

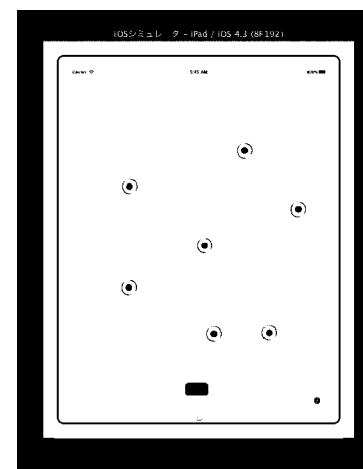


図5 認証画面

4.2.1 認証パス再現の検証

特徴点のみ表示することが、セキュリティとしてどれくらいの強度を持つのかを検証するために、著者の作成した認証パスを特徴点のみから再現できるのか試してみた。検証人数は10人それぞれに検証を行った。検証に使用するパターンは、次のABCの3パターンである。

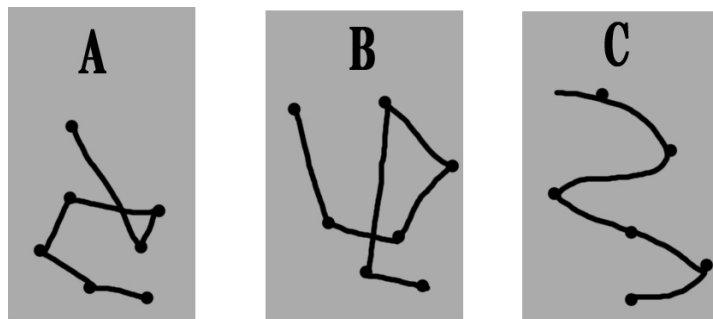


図6 認証パス再現検証パターン

表2 認証パス再現検証実験結果

	A	B	C
再現成功[人]	0	0	0
始点当のみ当たり[人]	0	0	2

結果として特徴点のみから認証パスを再現できた回数は0回だった。同じ検証を認証パスを変えて行ったが、結局再現出来たことはなく、結線開始の点を当てる事も非常に難しい事がわかった。つまり他人が認証オブジェクトのみから認証パスを再現する事はほぼ不可能だと考えることができる。

しかし万が一認証パスの再現が行えてしまった場合、或いは予め結線パターンが他人に知られてしまった場合は筆速の生体認証に頼る事になる。今後、筆速による生体認証を組み込んで、認証実験をおこなっていきたいと考えている。

現在、認証アプリケーションの開発は特徴点を取得し、認証パスを作る機能までが作成済みであり、これからは認証パスを複数人分端末に登録する機能と実際に生体認証を行う部分を作成しなければならない。

5. まとめと今後の課題

本論文では、携帯端末のタッチパネルを用いて、複数のから構成される図形を描画する際の筆速を用いて生体認証を行う方法を提案し、自由に描写された曲線から特徴点を取得し、そのパターンと筆速を用いて生体認証を行う方式を提案した。結果とし

て特徴点の自動認識は利用者に限り元の曲線の再現が行える程度の検出を行う事が出来ており、現在のところ、認証パスのパターンによる認証システムに用いる事に問題はないと考えられる。今後、このシステムに筆速による生体認証を組み込み、より、認証システムとしての強度を高めていきたいと考えている。

しかし、問題として残るのはこの自動検出システム自体が、生体認証に有益であるかは、実際に検証してみなければ判らないという事であり、検証結果次第ではシステムの大幅な改良を行う必要がある。ただ、現段階で認証パスの設定において、はユーザーが使用しやすくなるという点のみを見ても、このシステムには意味があると考えられる。従って、実際に生体認証システムを組み合わせ精度を検証しながら、アプリケーションを組み上げる事が今後の第1の目標となる。

参考文献

- 1) <http://www.android.com/>
- 2) 堂園 浩，中國真教他：自己組織化マップによる PDA における手書き図形の筆圧解析，情報処理学会論文誌，Vol. 47，No. 8，pp. 2593-2602 (2006)
- 3) Bolle, R., Connell, J., Pankanti, S., Ratha, N. and Senior, A.: Guide to Biometrics, Springer (2004)