Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

タッチパネルと加速度センサを用いた携帯端末向けジェスチャ認証とその入力方式の提案

見 上 一 憲[†1]　林 原 尚 浩[†1]

近年，タッチパッドや加速度センサなどを備えた携帯端末 (スマートフォンなど) の普及にともなって，携帯端末における認証方式も様々なものが提供されている．このような認証方式は，携帯端末特有の使用環境やデバイスの特性を考慮しなければならない．本稿では，タッチパネルと加速度センサを用いたジェスチャ認証とその入力方式の提案及び考察を行う．

Input Method of Gesture Authentication for Mobile Devices using Touch Panel and Accelerometer

Kazunori KENJO[†1] and Naohiro HAYASHIBARA[†1]

Recently, there are various kinds of mobile devices with a touch panel and an accelerometer (e.g., smart phones, tablet computers). Since they have rich function almost same as personal computers, various types of data can be stored and manipulated on them. Therefore diverse authentication methods have been developing in last few years. Important things on them are to take into account of the feature of mobile devices (i.e., mostly used in common space, etc.). In this paper, we propose an authentication method using a touch panel and an accelerometer. It can realize both an easy-input method and difficulty on understanding the gesture sequence by peeking.

1. はじめに

最近の数年間におけるスマートフォンの普及は著しく，携帯電話の総出荷台数のうちスマートフォンが占める割合は，2010 年度において 22.7%だったが，2011 年度は 49%へ大幅に伸び，2012 年度にはスマートフォンが過半数を占めると予測されている．[1]．スマートフォンなどの携帯端末は，従来のキーボードからパスワードを入力する認証方式を備えているものが多い．

しかし，この様な端末はパーソナル・コンピュータのように十分なキーピッチをもつキーボードを装備することは端末の特性上困難であり，パスワード認証をキーボードから行うことはユーザの利便性を損なうことが多い．このため，タッチパネルを用いて指でなぞった軌跡による認証や指紋認証，内蔵のカメラによる顔画像認証，加速度センサを用いた認証などが開発されている．これらの認証は携帯端末において，認証を行う際の利便性向上を目的としている．つまり，携帯端末向けの認証方式は，その入力方式の容易性も重要な要素である．

また，携帯端末はオフィスや自宅で使用するというよりも，外出時に持ち歩くケースが多い．特にスマートフォンは日常的に持ち歩き，屋内外問わず様々な場所で使用する．従って，周りに人がいる環境で操作する状況もある．このような場合，携帯端末自体もしくは，特定のファイルやアプリケーションの操作を認証を通して行う時に他人に見られてしまう可能性がある．

本研究では，携帯端末のタッチパネルと加速度センサを用いたジェスチャ認証方式を提案する．この認証方式はユーザの利便性を損なわずに，他人によるなりすましを困難にすることを目的としている．そのため，小型の携帯端末であれば片手で持った状態で容易に認証を行うことができるように考慮している．また，シンプルな動作の複雑な組み合わせによって，公共の場で他人に認証を行う動作を見られる状況にあったとしても，その動作の認識が困難な入力方式を提案する．具体的には，キーとパスワードによって構成される認証動作によって，同一のパスワードであったとしても，キーの設定やパスワードと無意味な動作とのインターリーブなどによる複数の動作パターンで認証を行うことができる．

2. 関連研究

近年，加速度センサを内蔵した携帯端末が多く開発され，それを用いた

†1 京都産業大学 コンピュータ理工学部
Faculty of Computer Science and Engineering, Kyoto Sangyo University

Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

認証に関する研究も多く行われている．

石原らによって提案された 3D 動作認証方式は，3 軸加速度センサを搭載した携帯端末を任意の方向に連続して振り，その動作軌跡を個人認証として用いるものである [2]．この認証方式は，加速度データをサンプリングし，それを基に動作区間を抽出する．これによって得られた動作区間の系列をジェスチャ (マスタデータ) として，認証時に得たジェスチャと比較して認証を行う．

この方式は，筆跡認証などと同様で，精度の高い認証を行うためには，定期的なマスタデータの更新が必要である．実験結果として 93%という高精度なジェスチャ認証を実現しているが，マスタデータの更新という余分な作業を伴うことが問題点としてあげられる．

Kirschnick らによる携帯端末用ジェスチャ認証の研究 [3] においては，17 人の被験者から入力された 11 種類，5,610 個のジェスチャデータを 5 種類の異なるジェスチャ認証判定方法を用いて測定している．その結果，8 の字を描くような”angular eight”というパターンを Parzen window classifier とサポートベクタマシーンを用いて認証を行った場合，99.94%の精度で認識できることが報告されている．

これら [2, 3] の研究では，ジェスチャの認識精度の実験を行っており，何れも特定の条件においては高い精度で認証が行えることを示している．しかし，これらは対象とするジェスチャが図形や文字を空間に書くような複雑なものであるため，認証を行うための計算量が多く，しかも，何度も試行することで，一見した他人でも認証を成功させることが出来る可能性が高い．また，[2] の実験結果によると，登録するジェスチャによっては容易になりすましが可能であるという結果も出ている．

一方，タッチパネルを用いたジェスチャ認証方式については，親里 (京セラミタ株式会社) によって出願された特許において提案されている [4]．従来，タッチパネルやタッチペンによって描かれた図形の比較照合は精度に問題があったが，入力可能領域に一定の判定線を設け，入力された図形がその判定線をどのように横切ったかという情報に基づいて認証を行うことで，認証精度の向上を図っている．

また，CGENE 社から Android OS によって動作する携帯端末向けの認証アプリケーション”Secret App Lock”においては，タッチパネルを使った認証を実装している [5]．この認証方式は特定のパターンの指の動作軌跡を登録して認証を行うものだが，前述の加速度センサを用いたジェスチャ認証と同様に，登録された動作軌跡と入力されたものを比較するものである．従って，しきい値の設定によっては，本人でも認証失敗する確率が高くなったり，誰でも認証できるようになったりする．つまり，しきい値の設定が非常に困難であることが問題点として挙げられる．

このアプリケーションにはタッチパネルを用いたジェスチャ認証も実装されており，タッチパネル上に入力した図形をもとに，認証時に入力された図形を近似して認証を行う．これもジェスチャ認証と同様の問題点が存在する．

2.1 関連研究の考察

関連研究として挙げたジェスチャ認証およびアプリケーションは，任意の動作をマスタデータとして保持し，それに対して認証を行った動作と近似した結果，しきい値以下であれば認証成功とするものである．複雑な動作を登録しておけば，他人のなりすましを防げる可能性は高いが，認証システム側のその動作における認証精度は低下することが考えられる．一方，単純な動作にすると，他人に認証動作を見られた際に，他人によるなりすましが容易になる．これは，タッチパネルを用いたジェスチャ認証においても同じことが言える．

このことから，ユーザの利便性となりすましの困難さはトレードオフの関係にあるということが言える．ユーザの利便性を向上させるためには，たとえばタッチパネル上の特定の点タッチするだけの認証などのように簡単に入力できるものが良い．しかし，このような認証は，動作の組み合わせが少ないためなりすましの危険性を高める．一方，タッチパネル上のキーボードから従来のように英数字を組み合わせたパスワードを入力する方法は，パスワードが十分長い場合，他人に成りすまされる危険性は低いが，ユーザの利便性を大きく損なうことになる．

本研究では，携帯端末においてユーザの利便性となりすましの困難さを両立する認証方法の確立を目的にタッチパッドと加速度センサを用いたジェスチャ認証を提案する．

Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

3. タッチパネルと加速度センサを用いたジェスチャ認証

本研究で対象とする携帯端末は，タッチパネルと加速度センサを持っているものとする．このような携帯端末上で動作するジェスチャ認証システムを提案し，iPhone でのプロトタイプの実装を行う．

3.1 ジェスチャの定義

本研究では，タッチパネル上での指の動作と携帯端末を振ることによって加速度センサから得られる端末の移動軌跡をジェスチャと定義する．タッチパネルの動作と携帯を振る動作は，ジェスチャの定義によっては，どちらも認証動作としてシームレスに統合できる．また，これらの動作を組み合わせることによって，他人に認証動作を見られていたとしても，解読をより困難にすることができる．

ジェスチャは予め登録されているものとし，その登録されたジェスチャと入力されたジェスチャをマッチングさせることによって認証を行う．以下の節でタッチパネルを用いたジェスチャと加速度センサを用いたジェスチャについて定義を行い．実装において必要なジェスチャのシンボル化を行う．

3.1.1 タッチパネルを用いたジェスチャ

本システムでは，タッチパネルと加速度センサの両方を用いて認証を行う．そのため，携帯端末を振りやすいように持っていることを考慮しなければならない．つまり，タッチパネルのジェスチャが複雑で正確な図形などを描かなければならない場合，タッチパネルを目視するため携帯端末を動かす必要がある．しかし，この動作により加速度センサが反応して意図しないジェスチャと認識する可能性がある．このことを考慮して，タッチパネル上でのジェスチャは目視しなくても十分に入力できる動作でなければならない．

本システムでジェスチャとして認証に用いるタッチパネル上のジェスチャを以下に定義する．

3.1.1.1 スワイプ

スワイプはタッチパネル上に一本の指で直線を描くように移動させる動作である．厳密にはタッチパネルに触れた座標から離した座標までの移動距離において，x 軸，y 軸方向の長い方を移動方向の情報として抽出し，

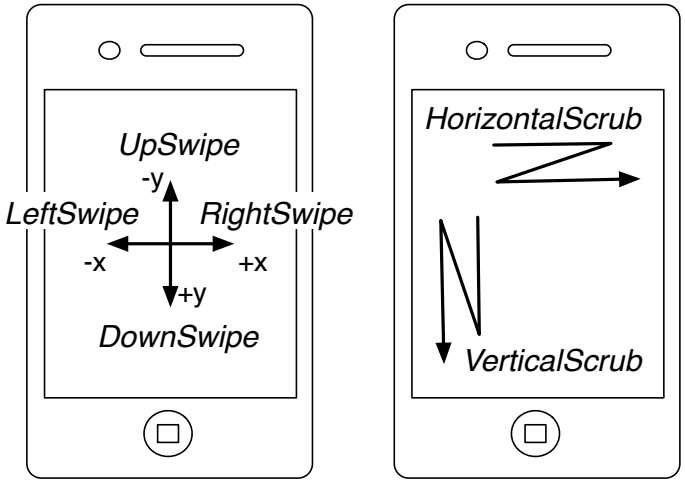

図 1 タッチパネルを用いたジェスチャ
Fig. 1 Gestures using touch panel

移動距離が設定されたしきい値より大きい場合のみスワイプとして認識される．図 1 に示すとおり，スワイプは以下の 4 種類を対象としている．

UpSwipe x 軸の負方向へ指を動かす
DownSwipe x 軸の正方向へ指を動かす
LeftSwipe y 軸の負方向へ指を動かす
RightSwipe y 軸の正方向へ指を動かす

3.1.1.2 スクラブ

スクラブはタッチパネル上に一本の指でこする動作を指す．これは，タッチパネルから指を離さず，スワイプの動作を方向を変えて繰り返し行うものである．例えば，図 1 に示すように，右にスワイプした後，指をタッチパネルから離さずに左にスワイプし，それらを繰り返す動作がスクラブとして認識される．

本システムでは，以下の 2 種類のスクラブのみを対象としている (図 1 参照)．

HorizontalScrub 水平方向に指を動かす

Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

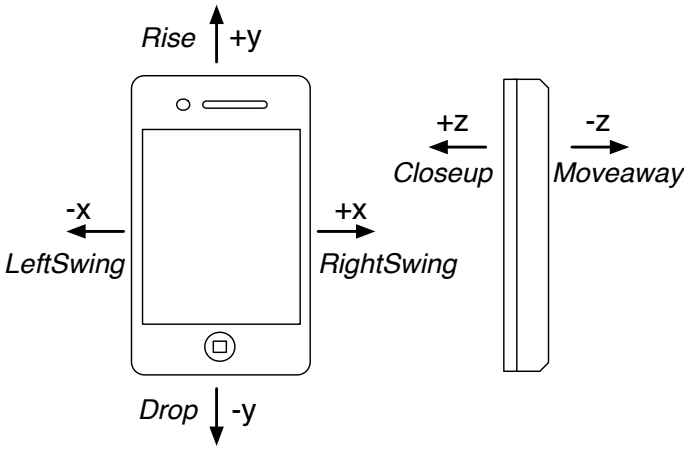

図 2 加速度センサを用いたジェスチャ
Fig. 2 Gestures using accelerometer

VerticalScrub 垂直方向に指を動かす

3.1.2 加速度センサを用いたジェスチャ

端末に内蔵されている加速度センサは図 2 に示すとおり 3 軸の加速度を検出出来ることを前提としている．

加速度センサを用いたジェスチャは図 2 に示すとおり，以下の 6 種類を定義する．

Rise 携帯端末を y 軸の正方向に振る
Drop 携帯端末を y 軸の負方向に振る
LeftSwing 携帯端末を x 軸の負方向に振る
RightSwing 携帯端末を x 軸の正方向に振る
Closeup 携帯端末を z 軸の正方向に振る
Moveaway 携帯端末を z 軸の負方向に振る

3.1.3 ジェスチャのシンボル化

ジェスチャ認証の実装に向けて，3.1 節で定義した各ジェスチャにシンボルを割り当てる．定義したジェスチャは，タッチパネルを用いたもの，加速度センサを用いたものがそれぞれ 6 種類あるので，シンボルが 12 個必要となる．本実装においては，12 種類のシンボルの集合 $Æ = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b\}$ を定義し，このシンボルにジェスチャを割り当てる．3.1.1 節と 3.1.2 節で定義したジェスチャとシンボルの割り当ては表 2 のとおりである．また，指および携帯端末の動きとシンボルとの対応関係は図 3 のようになる．

表 1 ジェスチャとシンボルの対応
表 2 Mapping between gestures and symbols

ジェスチャ	シンボル	ジェスチャ	シンボル
Rise	0	UpSwipe	6
Drop	1	DownSwipe	7
LeftSwing	2	LeftSwipe	8
RightSwing	3	RightSwipe	9
Closeup	4	HorizontalScrub	a
Moveaway	5	VerticalScrub	b

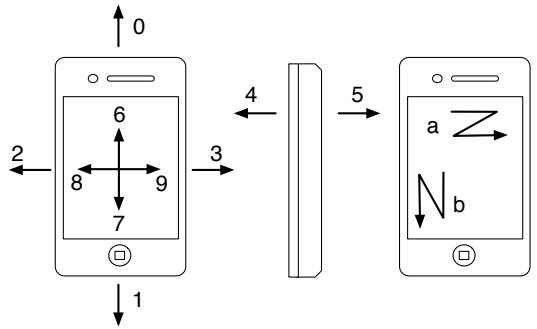

図 3 携帯端末の動きとシンボルの割り当て
Fig. 3 Mapping between movement of a mobile device and symbols

連続して入力された n 個のジェスチャはシンボル化された長さ n のジェスチャの系列 $G = (p_1, p_2, \ldots, p_n)^{\star 1}$ として表現される．以下では，G の

$\star 1$ ただし，$\forall p : p \in G \Rightarrow p \in Æ$

Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

ことをジェスチャパターンとよぶ.

4. ジェスチャ認証のための入力方式

携帯端末の使用環境を考慮すると，自宅やオフィスなどの閉鎖的な空間よりも，屋外などの公共の場での使用を想定しなければならない.

従って携帯端末の認証システムに関しては，このような環境下で起こりうるユーザの不利益を想定する必要がある．つまり，他人に見られている状況で認証を行い，かつ，どのような認証を行なっているかが他人には分かりにくいものでなければならない.

本研究で提案する認証システムは，キーとパスワードの組合せによって認証を行う．キーとパスワードはそれぞれ前節で定義したジェスチャを用いる．キーは 1 つのジェスチャ$k \in Æ$，パスワードは n 個のジェスチャの組み合わせによって定義される．従ってパスワードは，n 桁のジェスチャパターン $G = (p_1, p_2, ..., p_n)$ から構成される．キー k とパスワードとなるジェスチャパターン G は予めユーザによって登録されるものとする.

キーとパスワードとなるジェスチャパターンの入力を説明するために，2 種類のシンボル $\alpha \in \{x | x \in Æ - \{k\}\}$，$\beta_i \in \{y | y \in Æ - \{k, p_i\}\}$ (ただし $0 < i < n$) を導入する．つまり，α はキー k 以外の任意のジェスチャ，β_i は α の取り得るジェスチャから p_i を除いたジェスチャを指す．α，β_i は認証には関係の無いジェスチャであり，これらをダミージェスチャと呼ぶことにする.

キーの役割は，キーに続くジェスチャの系列をパスワードと認識させる役割がある．キーの後に入力されたジェスチャの系列が予め登録されているパスワード G と一致すれば，認証成功となる．また，キーに続いて連続してジェスチャを入力している際，i 個目のパスワードは p_i を入力しなければならないが，異なるジェスチャβ_i が現れた場合は p_{i-1} までをパスワードと認識し，それ以降は再び k が入力されるまではパスワードと認識しない．つまり，パスワードとしてジェスチャを認識させるためには，そのパスワードの前に接頭辞として k を入力する必要がある．一方，k が入力された系列に現れないときは，パスワードが全く入力されていないということになる.

ジェスチャー認証において認証が成功するジェスチャパターンの系列は

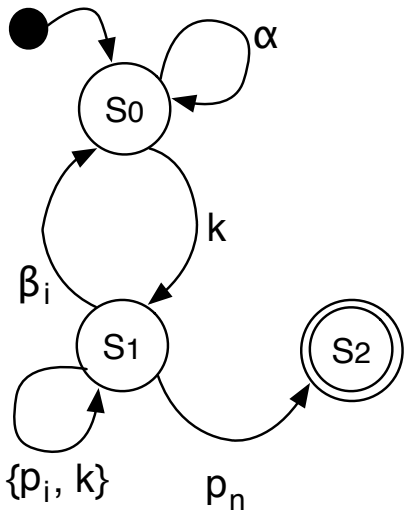

図 4 パスワードが n 桁の場合の状態遷移図
Fig. 4 DFA in the case of n-figures password

図 4 の有限オートマトンで表すことが出来る.

図 4 の p_i^* は $p_1 p_2 \dots p_{n-1}$ というジェスチャの系列を示す．n 桁のパスワードを入力する場合，一番単純なパスワード入力方法としては，$kp_1p_2 \dots p_n$ という系列でジェスチャを入力する方法が挙げられる．この場合は，k に続く $p_1 \dots p_n$ がパスワードと認識される.

これ以外にも，図 4 のオートマトンは $\alpha^* k^* p_i^* p_n$ のようにキー k を入力する前にキー以外の任意の動作 α をが現れる系列や，$k^* p_1 \beta_2 \alpha^* k^* p_2 p_3 \dots p_n$ のようにパスワード入力中に α や β_i が現れる系列でも受理する (つまり，認証成功となる．)．これは，再びキー k が入力された場合はそれに続くジェスチャはパスワードの続きとして認識され，最終的に $p_1 p_2 \dots p_n$ の順でパスワードとして入力されているためである.

図 4 で示したオートマトンで受理可能な系列を正規表現で表したものが式 (1) である.

$$((k^* + \alpha^* k^*)(p_i^* + p_i^* \beta_{i+1} + \beta_i))^* (p_n + k^* p_n) \quad (1)$$

Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

5. iPhone を用いたプロトタイプの実装

現在，タッチパネルと加速度センサを持つ携帯端末 (スマートフォン含む) は様々な種類が存在する．本研究では，提案するジェスチャ認証システムのプロトタイプを Apple 社の iPhone 4S (OS: iOS version 5.0.1) を用いて実装を行った．開発環境は Mac OS X Lion (version 10.7.2) 上で Xcode version 4.2.1 を用い，実装には Objective-C を用いて行った．

5.1 ジェスチャの取得

iOS 5 においてタッチパネルと加速度センサを用いたジェスチャを取得するために Xcode 4.2.1 に付属している iOS SDK を用いている．以下に記載するクラスは iOS SDK に含まれるクラスである．

5.1.1 タッチパネルを用いたジェスチャの取得

スクリーン上の指の位置，運動などを `UIEvent` クラスを通してイベントとして受け取り，それに対応した処理を行う．タッチパネル上のジェスチャの認識は，指がタッチスクリーンに触れた際のイベント `UITouchPhaseBegan` から指がタッチスクリーンはら離れた際のイベント `UITouchPhaseEnded` の間の指の移動距離，方向を検出することで行う．誤操作を防止するために，しきい値を設定しており，指の移動距離がしきい値以下の場合は，ジェスチャと認識しない．

スワイプは，タッチパネル上の 4 方向 ($\pm x$，$\pm y$) の指の移動から構成される．指が斜めに移動した場合などは，移動距離が長い方の軸を方向情報として取得し，対応するジェスチャとして認識する．

タッチスクリーンから指が離れずに移動する方向が逆方向に 2 回以上変更されれば，スクラブと認識する．

5.1.2 加速度センサを用いたジェスチャの取得

`iOS SDK` では，`iPhone` に搭載されているオンボードの加速度センサから 3 軸方向の加速度の値をそれぞれ取得するためのクラス `UIAccelerometer` が提供されている．このクラスをインスタンス化し，プロパティ `delegate` によって，更新された加速度の値を提供するオブジェクトを指定する．

加速度センサを用いたジェスチャは基本的な 6 方向 ($\pm x$，$\pm y$，$\pm z$) の動作から構成されている．そのため，ジェスチャを行った際に最も大きい加速度の値を持つ軸の値を方向情報として認識し，対応するジェスチャに変換する．加速度センサを用いたジェスチャにおいても，しきい値を設けており，加速度の値がしきい値以上になった場合にのみジェスチャとして認識する．

6. 考　　察

前節では，携帯端末のジェスチャ認証のために，キーとパスワードの組み合わせで複数のジェスチャパターンによる認証を可能とする入力方式を提案した．この入力方式により，従来の単一のジェスチャによる認証よりも複雑になり，ジェスチャ認証を行っている際に他人に見られたとしても，キーとパスワードを見破ることが容易ではなくなる．本節では，実際にどれだけのジェスチャパターンが存在するかについて議論する．

3 桁のパスワード $G = (p_1, p_2, p_3)$ とキー k について考える．α と β_i が現れない場合，式 (1) より以下のジェスチャパターンが考えられる．

$$k^* p_1 p_2 p_3 + k^* p_1 p_2 k^* p3 + k^* p_1 k^* p_2 p_3 + k^* p_1 k^* p_2 k^* p_3 \qquad (2)$$

これは，キーとパスワードの組み合わせを変えることなく，α や β_i のようなダミージェスチャを入れなくても，4 種類の異なるジェスチャパターンによる認証を行うことができることを示している．

このジェスチャパターンの組み合わせ総数（ジェスチャパターン数）とパスワードの桁数 n の関係は，n 回サイコロを振ったときの和が n となる場合の数に帰着して考えることができる．ただし，サイコロの面の数は 0〜n の $n+1$ である必要がある．$n = 3$ の例で考えると，式 (2) における 4 種類のジェスチャパターンは $(3,0,0)$，$(2,1,0)$，$(1,2,0)$，$(1,1,1)$ と一致する．このとき，$\{(0,3,0),(0,0,3),(3,0,0)\}$，$\{(0,2,1),(2,0,1),(2,1,0)\}$，$\{0,1,2),(1,0,2),(1,2,0)\}$ のように 0 から始まるパターンや 0 が 0 以外の数字とインターリーブするパターンは，0 以外の数字が同じ順序で現れる他のパターンと同一と見なす．

従って，パスワードの桁数とジェスチャパターン数は表 3 のようになる．

表 3 パスワード桁数に対するジェスチャパターン数

パスワード桁数	1	2	3	4	5	6	7	8
ジェスチャパターン数	1	2	4	8	16	32	64	128

Vol.2012-DPS-150 No.8
Vol.2012-CSEC-56 No.8
2012/2/29

表 3 により，α や β_i を含まない場合でもジェスチャパターン数はパスワードの桁数 n に対して 2^{n-1} となることが分かる．さらに，α や β_i のダミージェスチャを含むようなジェスチャパターンを考えるとこれ以上になり，それらを上手く組み合わせることによって，パスワードやキーを変更しなくても，他人が見て解読が困難であるような認証動作を実現することができる．もちろん，キーやパスワードを定期的に変更することによる，セキュリティ面の自己防衛も必要である．

7. まとめと展望

本稿では，加速度センサとタッチパネルを用いた携帯端末向けのジェスチャ認証システムを提案した．このシステムは，ユーザの利便性を損なわずに，他人によるなりすましも困難にすることを目的とした認証システムである．そのため，ひとつひとつのジェスチャはシンプルな動作であり，入力も容易であるが，認証を行うためにはそれらの複雑に組み合わせることが可能である．つまり，同一のキーとパスワードであってもパスワードの桁数によっては複数の認証可能なジェスチャパターンが存在するため，認証中に他人が見ていたとしてもそれらを解読することが容易ではないような認証動作を実現することができる．

提案システムのプロトタイプを iPhone に実装しているが，ジェスチャ認識の精度，ユーザから見た使いやすさ，なりすましの困難さについては，今後，実験を行って検証していく必要がある．また，入力できるジェスチャの種類を増やすことによって，認証システムの強度を向上させる予定である．

参 考 文 献

1) 基本戦略ボード事務局，"我が国の ICT に関する現状と動向"，総務省 情報通信審議会 (2011)．
2) 石原，太田，行方，水野，"端末自体の動きを用いた携帯端末向け個人認証"，情報処理学会論文誌，Vol. 46，No. 12，2005．
3) N. Kirschnick, S. Kratz and S. Möller, "An Improved Approach to Gesture-Based Authentication for Mobile Devices", In Proc. of Symposium on Usable Privacy and Security(SOUPS) 2010, July 2010, WA, USA.
4) 親里，"ジェスチャー認証方法及びジェスチャー認証装置"，特許，公開番号：2008171066，2008
5) 有限会社 CGENE，"Secret App Lock"，http://android.specialist.jp/secret/applock/