

## 口唇領域の動きの画像認識を用いたスマートデバイス向けパターンロックシステム Pattern-Lock System for Smart Devices by Using Motion Recognition of Lip Area

岸里 正樹<sup>†</sup>      高井 昌彰<sup>‡</sup>  
Masaki Kishisato      Yoshiaki Takai

### 1. はじめに

現在、スマートフォンやタブレット端末など、タッチスクリーンを搭載したスマートデバイスでは、個人認証の方法の一つとしてパターンロック認証が用いられている[1]。しかし、手の指によって行うパターンロック認証は、肩越しに画面を覗き見るショルダーハックや、画面に残った脂の跡を読み取ることによって、認証キーを盗まれてしまうリスクを持つ。そこで手の指の代わりに、口唇領域の動きを端末のカメラ画像から認識することで認証を行うパターンロック認証システムを開発した。これは、視線や瞬きは拘束せずに頭部を動かさない状態で、自由度の高い可動部位は口唇領域のみであることを応用したものである。これにより、画面に触れることなく、手の指を用いる場合と近い感覚で認証処理を実現できる。

### 2. パターンロック認証について

パターンロック認証とは、図1に示すように、複数個の点をなぞった順番をキーとして認証を行うものである。図1の場合、認証キーとして入力されたパターンを割り当てられた数字の列に置き換えると“12765483”となる。

パターンロック認証を行うことで、従来の認証方法によって得られるものよりもはるかに多い組み合わせ数の中から認証キーを作成することができる。



図1 パターンロック認証の例

### 3. 本システムの概要

本システムの流れを図2に示す。

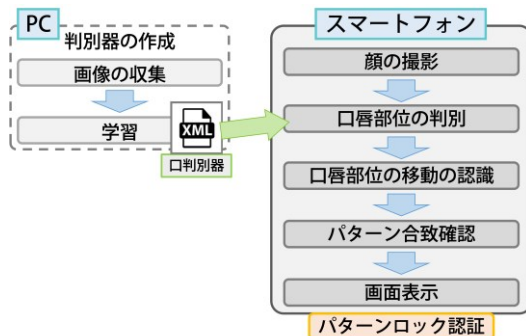


図2 システム概要

<sup>†</sup> 北海道大学大学院情報科学研究科, Graduate school of Information Science and Technology, Hokkaido University

<sup>‡</sup> 北海道大学情報基盤センター, Information Initiative Center, Hokkaido University

### 3.1 判別器の作成

カメラ画像から人物の口唇領域の座標を取得するための判別器を OpenCV で作成する。

まず、判別対象である口唇部位を含むポジティブ画像、口唇部位を一切含まないネガティブ画像のファイルをそれぞれ用意し、OpenCV の学習用プログラムに読み込ませ、学習を行う[2]。学習完了後、XML 形式のファイルとして生成された判別器をスマートフォン内に取り込む。本システムで用いる判別器は、画像から Haar-Like 特徴量を取得して、口唇領域の判別を行うものである。

### 3.2 パターンロック認証

#### 3.2.1 パターンロックの仕様

本システムで用いるパターンロックは、以下に列挙するルールに基づいている(図4参照)。本稿ではパターンの個々の構成要素(隣接格子点間の移動)を move と呼ぶ。

- 4 × 4 の格子点上を移動
- 上下左右4方向への移動が可能
- 一つの格子点を複数回通ってもよい
- 隣接格子点を結ぶ線を複数回通ってはならない
- 初期位置は既定のスタート点に固定

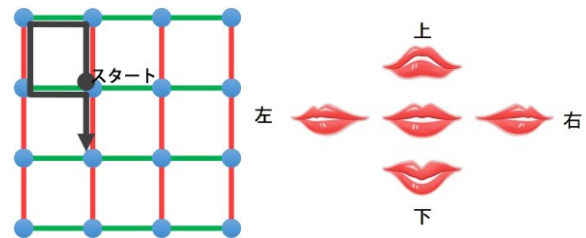


図4 本システムで用いるパターンと操作方法

#### 3.2.2 口唇部位の判別

顔の撮影画像から口唇部位を含む矩形領域(口唇領域)を抽出するため、カメラ画像の左上から毎フレーム走査を行い、右下まで到達したら走査領域を拡大し、再び左上から走査を繰り返す。走査中に口唇部位が認識されると、その都度口唇領域を保存し、走査領域が最大に達した段階で走査を終了する。

走査の結果、複数回口唇部位を認識することがあるが、その場合は重なっている口唇領域を同一化する処理を行う。

#### 3.2.3 口唇部位移動の認識

過去8フレーム分の口唇領域の重心座標を保存し、毎フレームの重心座標と過去の平均の重心座標との差を計算する。x成分・y成分のどちらかが閾値を超えた場合、ある方向へ口唇部位が移動し、パターンの入力となされたものとみなす。

### 3.2.4 移動パターンの認識

口唇部位移動の認識をトリガーとして、方向入力の際に応じて、移動パターンを検出する。方向入力がある一定フレーム数継続した場合に、入力パターンに Move を追加する。移動パターンのルールに反する Move の入力は無視する。

一度 Move が追加されると、次の Move を行う前に停止待ち状態に遷移し、口唇部位が一定フレーム数停止したことを確認してから次の Move を受け付ける。

一定の停止時間を設けることで、口を戻す動作を認識して意図せぬ入力がおこることを防ぎつつ、次の Move を待つことができる。

### 3.2.5 移動パターンの合致確認

新たにパターンの入力が行われる度に、これまで入力されたパターンと予めシステムに設定された正解パターンとの照合確認を行う。

合致した場合、パターンロックが解除されたと判定し、処理を終える。合致しなかった場合、以降どの方向にも移動できなくなった場合を除き、入力の受付を継続する。

正解パターンと合致せず、かつ 4×4 格子パターン上で通過可能な線がもう存在しない場合、ロック解除失敗として処理を終える。

## 3.3 システムの実装とパフォーマンス

本システムの実装には、判別器作成用端末として、CPU : Intel Xeon CPU E5645, メモリ : 12.00GB, OS : Windows 7 Professional SP1 64bit の PC を使用した。開発言語として Java, 画像処理ライブラリとして、OpenCV を使用した。また、パターンロック認証を行うスマートフォンとして、GALAXY Note II SC-02E を使用した。

## 4. 動作結果

### 4.1 判別器の作成

ポジティブ画像約 750 枚、ネガティブ画像約 550 枚、計 1300 枚程度の画像を収集した。ポジティブ画像にはウェブカメラで人物の顔を撮影して作成した連続写真や検索エンジンで得られた顔画像を用いた。ネガティブ画像には風景の写真やアニメのキャラクターの画像などを用いた。

収集した画像を用いて学習を行った結果、判別器作成完了までに約 2 時間を要した。作成された判別器のファイルサイズは約 75KB である。

### 4.2 動作画面

スマートフォンの描画画面には必要最低限の情報のみを表示する。動作中の実際の画面及び操作者の様子を図 5 に示す。

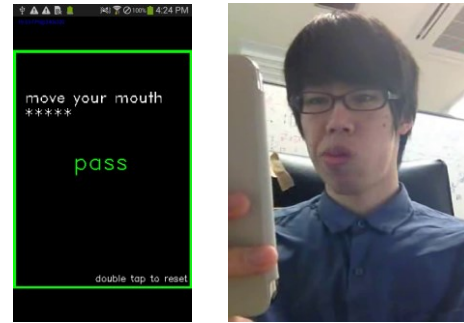


図 5 操作画面及び操作者

### 4.3 入力時間・Move 数

Move 数が 4~9 であるパターンをそれぞれ用意し、10 回ずつ入力を行い、パターンロック解除までに要した時間を計測した。その平均時間の代表的な結果を表 1 示す。

Move 数	パターン	平均入力時間 (s)
5	↑→↓↓	1.8101
5	↑←↓→↓	1.8956
7	←↓→→→↑↑	2.8292
9	↑←↓→↓←↓→↑	4.0024

表 1 パターン別入力時間の平均

パターンの Move 数の増加に伴い、入力時間も増加している。Move 数 5 の場合に示したように、“↓↓↓”といった同じ方向への連続した入力を含むパターンの方が、そうでないものよりも短い時間で入力を行っている。これは、口唇部位を繰り返し同じ方向へ移動させる方が速く入力が行えるということが理由として考えられる。

また、最も長い Move 数 9 のパターンを用いても 4 秒程度で入力を行えるため、スマートフォンのパターンロック認証システムとして日常的に利用する際にも実用に堪えることが考えられる。

## 5. まとめと今後の課題

ショルダーハック等による認証キーの盗難のリスクを回避するため、指の代わりに口唇部位の移動によって操作を行うパターンロック認証システムの開発を行った。

今後の課題として、入力精度の向上、口の開閉や舌の移動などによる操作の実現、口唇部位認識のパターンロック認証以外への応用が挙げられる。

### 参考文献

- [1] 石黒司, 福島和英, 清本晋作, 三宅優: “モバイル端末のロック解除向けパターン認証の安全性評価”, 情報処理学会研究報告, Vol.2012-CSEC-58, No.41, 2012.
- [2] 技術評論社: “第 3 回オブジェクト検出してみよう : OpenCV で学ぶ画像認識”, 技術評論社  
<http://gihyo.jp/dev/feature/01/opencv/0003?page=2>  
 (参照 2015 年 6 月 19 日) .