

指間の線を利用した掌紋認証の提案

上松 晴信^{†1} 神田 龍一^{†1} 松井 利樹^{†1} 三宅 優^{†1}
伊藤 康一^{†2} 青木 孝文^{†2}

概要: スマートフォンにて他者による不正使用を防止するために生体認証機能の搭載された端末が増えており、個人的特徴の1つとして手のひらを使用した認証が提案されている。手のひらを使用した認証では、手のひらの輪郭を認証に使用する方法や、掌紋と呼ばれる手のひらの領域を認証に使用する方法がある。既存研究では、手のひらを認識する条件としてある一定の環境下、つまり背景が単一色で他の物体が映っていない環境で撮影されたものに限定している場合が多い。スマートフォンなどに搭載されているカメラで認証を行う際は、撮影環境が一定でない。そのため、背景に肌色の物体や複数の物体が映っていることが多く、その中で手のひらを抽出することは容易ではない。本論文では手のひらをかざした時にできる指間の線に着目した掌紋認証を提案する。本方式では、指間の線を正確に抽出し手のひらの角度および縮小率を計算することで、手のひらの回転および拡大・縮小に対応している。そのため、認証を行うユーザは手のひらの向きおよびカメラからの距離を意識することなく認証することができる。本提案方式の有効性を示すため532名のデータセットを使用して評価を行った。

キーワード: 掌紋認証, 画像処理, パターン認識

The proposal of palmprint authentication using the lines between the fingers

Harunobu Agematsu^{†1}, Ryuichi Kanda^{†1}, Toshiki Matsui^{†1}, Yutaka Miyake^{†1}
Koichi Ito^{†2}, Takafumi Aoki^{†2}

Abstract: Biometric authentication using a hand has been proposed to prevent the unauthorized use of mobile phones by another person. There are some types of hand biometrics such as a hand shape and a palmprint. In the previous works, hand images have to be acquired under the limited conditions where the background color is uniform and other object is not included in the photo. On the other hand, it is difficult to prepare such stable conditions for image acquisition when using a camera embedded in smartphones. Thus, it is difficult to extract a palm region from an image including multiple objects with the complex background. Addressing this problem, we propose a palmprint authentication method using lines between fingers. Our method extracts lines between fingers and calculates rotation angles and scale factors of them. Therefore, user authentication can be done without regard to hand pose changes and the distance from a camera to a hand. We evaluate performance of the proposed method using our dataset having 532 users to demonstrate its effectiveness.

Keywords: Palm Print Authentication, Image Processing, Pattern Recognition.

1. はじめに

近年、スマートフォンを使用したオンラインショッピングや電子マネーによる決済が日常的に行われるようになった。それに伴ってスマートフォン内に個人情報などの重要なデータが保存されるようになってきている。それらのデータを守るために様々なセキュリティ機能がスマートフォンに搭載されている。一般的に利用されている個人認証機能としては、パスワードや PIN (Personal Identification Number) などがある。しかしこれらの認証は、記憶による認証のため、他者によって盗まれたり、本人が忘れてしまったりする問題がある。

人間の特徴を利用した認証スキームであるバイオメトリ

クスは、コンピュータビジョンやパターン認識などの多くの技術から成り立つ。バイオメトリクスで使用される個人的特徴は、顔、指紋、虹彩、網膜、手や指の幾何学模様などの身体的特徴および歩き方、声、署名およびキーストロークなどの行動的特徴などがある。適用先のセキュリティレベルにより、これらの個人的特徴を2つ以上組み合わせるマルチモーダルスキーム [1] が提案されている。顔認証は、アクセサリや表情、姿勢、光の当たり具合などに敏感である。虹彩や網膜を使用する認証では、特別な照明が必要であり、ユーザビリティが低い。指紋認証では、指紋の隆線が不鮮明なために指紋で認証できない人が4%程度も存在するとされている [1]。数多くの特徴の中で、

^{†1} 株式会社 KDDI 研究所
KDDI R&D Laboratories, Inc.

^{†2} 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

手のひらを使った個人認証は、デジタルカメラやスマートフォンがあれば簡単に手のひらの画像を撮影できるため、特別な装置を用意する必要が無いという利点である。

非接触型掌紋認証の従来研究では、スマートフォンのカメラで撮影した手のひら画像から掌紋領域を抽出する前処理においてキーポイントと呼ばれる指と指の間のくぼみを自動的に検出することが難しいという課題があった[2]。そのため、認証できるようにユーザが手のひらの位置を合わせる必要があり、登録および認証に時間がかかっていた。本論文では、手のひら認識技術を応用することでユーザによる位置合わせを必要としない非接触型掌紋認証を提案し、その有効性を評価した。

2. 関連研究

文献 [3] では、手の輪郭の形を個人的特徴として認証を行っている。手のひら画像に対して、主成分分析を行い、重心位置および主軸角度を算出することで、手のひらを正規化している。具体的には、まず、K-means クラスタリングを用いて手のひらと背景に分割する。その後、抽出した手のひらの輪郭線に対して、手首の中心点から輪郭上の点までの距離を算出する(図1)。極小値として得られる指間のくぼみに基づいて5本の指を切り出し、それぞれに対して主成分分析を行い、重心および主軸角度を決定する。主軸角度を基準角度に合わせてそれぞれの指を回転させることで正規化(位置の補正)を行う。この方法により、ユーザは、手のひらおよび指の方向を気にすることなく認証を行うことができる。提案されている方法では、指輪や腕時計などの人工物を装着している場合でも方向を正規化することができる。ただし、入力画像を手のひらと背景に分割できることを前提としている。つまり、限定された環境下で手のひら画像を撮影する必要がある。

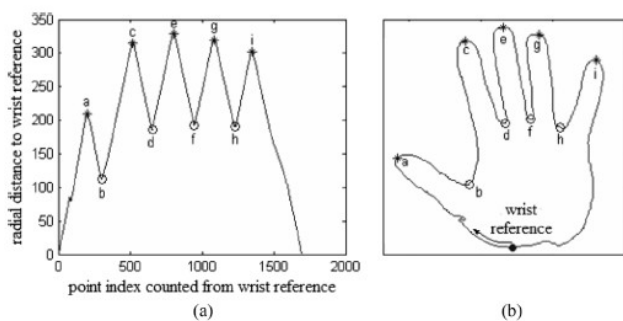


図1. (a)距離グラフ,(b)手のひらの輪郭

Figure 1. (a)Radial distance function for finger extraction and (b) a hand contour with marked extremities.

携帯電話やスマートフォンで認証を行う場合は、撮影環境が一定でない。つまり、背景に肌色の物体や複数の物体が映っていることが多く、その中で手のひらのみを抽出す

ることは容易ではない。加えて、照明の変化にもある程度対応する必要がある。文献 [4] では、撮影環境が一定でない場所で手のひらを抽出するための方法を提案している。肌色とみなされた画素を初期点として、領域成長法を使用して手のひらの領域を抽出している。抽出した手のひら画像に対して図1と同様な距離グラフを作成し、キーポイントの抽出を行っている。キーポイントの抽出時、複数の点が候補になることを考慮し、人差し指-中指、中指-薬指、薬指-小指のくぼみの位置関係が二等辺三角形になるような点をキーポイントとしている。

文献 [5] で提案されている手の検出手法では、さまざまなサイズのウィンドウで画像を走査し、検出ウィンドウ内の画像に対して手のひらの識別処理を行う。これにより、手の大きさの個人差や、カメラからの距離に関わらず手のひらを検出できる。各ウィンドウの識別処理では、まず、検出ウィンドウ内の画像を部分領域に分割し、各部分の領域内で輝度勾配方向(各画素に対して、より高い輝度の画素が存在する方向)および相対色(基準色に対する各画素の色差)を算出する。次に、各部分領域での輝度勾配方向の共起および相対色の各出現頻度からヒストグラムを生成し、特徴ベクトルとする。多数の正解および不正解サンプルの特徴ベクトルから、あらかじめ学習により求めておいた識別境界に基づき、どちらのサンプルに近いかを判定することで、手のひらかどうかを識別する。輝度勾配方向を使うことで、照明環境による輝度差変化にロバストな点、相対色を使うことで、照明による色成分の変化を抑え、安定して類似色パターンを抽出できる点、ヒストグラム化により手の形状や姿勢の個人差を吸収して安定に手のひら領域を抽出することができる点が強みである。この手法の課題は、様々なサイズのウィンドウを設定して画像を走査する必要があるため、手をかざしてから画面上で認識するまでに処理時間がかかる点である。また、ウィンドウ内で算出した特徴ベクトルを分類器にかけて手のひらかどうかを識別するため、膨大な数の手のひらのサンプルが必要となる。

3. 指間の線を利用した手のひら認識

本章では、指間の線を利用した手のひらの認識方法について述べる。

3.1 概要

提案する方式では、まず、入力画像に対してライン検出処理を行い、線分を抽出する。抽出された線分の中には指間の線ではない短いものが多く含まれるため、線分のフィルタリングを行い、ある一定の長さ以上のものを選択する。選択された複数本の線分の中から、人差し指、中指、薬指の間の三本線分を抽出する。ここで、抽出された三本線が指

の根元まで達していない場合があるため、線分同士の位置関係を確認し、条件に満たない場合は線分の補正処理を行う。その後、キーポイントから掌紋領域を切り取る処理を行う。この処理中に、検出された線の傾きおよび角度から手のひら全体の回転を補正する。また、手のひらの大きさからカメラと手のひらの距離を求め、縮小・拡大に対応する。



図 2. (a) グレースケール画像, (b) 平滑化した画像
Figure 2. (a) Grayscale image, (b) Smoothed image.

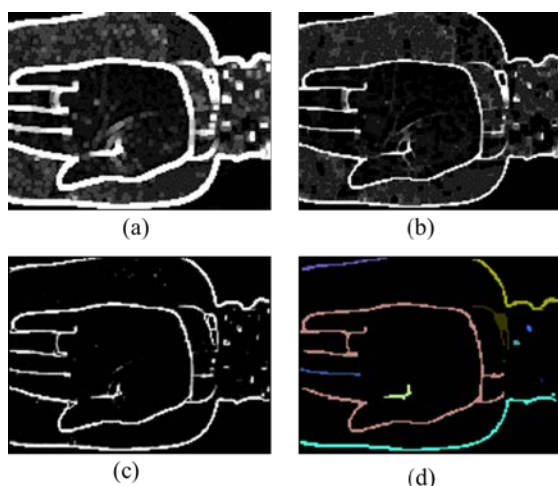


図 3. (a) 膨張, (b) 収縮, (c) ノイズ除去後の画像, (d) ラベリング

Figure 3. (a) Dilation, (b) Erosion, (c) Noise reduction image, (d) Labeling.



図 4. (a) ライン検出後の画像, (b) 指間の線を抽出した画像

Figure 4. (a) Line detected image, Lines between the fingers.

3.2 ライン検出処理

ライン検出処理では、初期ステップとして入力画像をグレースケール化し (図 2 (a)), ガウシアンオペレータを使用して平滑化を行う (図 2 (b)). ガウシアンオペレータは、正規分布にしたがって重みを定めた加重平均オペレータである。注目画素に近いほど重みが大きくなるオペレータである。注目画素の画素値に最も大きな重みをつけることで、単純な平均化オペレータと比べて、元の画素値を残すような平滑化が実現できる。

カメラからの入力画像は、画素値が一樣でなく、ムラがある。そのため、平滑化後の画像をそのまま二値化すると、得られる二値化画像には多くのノイズが入ってしまう。このようなノイズを除去するために、二値化画像に対して膨張処理および収縮処理を行う。膨張処理では画像内の図形を外側に一定画素分広げ (図 3(a)), 収縮処理では画像内の図形を内側に一定画素分狭める (図 3(b)). 膨張処理を複数回繰り返して実行することで、分割した領域を結合することができる。収縮処理を複数回実行することで、小さな独立領域を消去して最大領域のみ残すことができる。膨張および収縮を繰り返して実行することでノイズ除去する (図 3(c)). そして、ラベリング処理では、二値画像内に存在する一連のつながった要素に対して、一意の番号を付ける (図 3(d)). ラベリングされた点群から曲率が極大になる点を検出し、点と点の間の線分を抽出する (図 4(a)). 線分の長さが一定の値よりも短いものは、線分のリストから削除する。

3.3 ライン認識処理

ライン認識処理とは、選択された複数本の線分の中から、人差し指、中指、薬指の間の三本線を抽出する処理のことである。この処理を行うにあたって、RGB 色空間ではなく、YUV 色空間の情報を用いる。YUV 色空間とは、輝度信号 Y と 2 つの色差信号 U と V で表現される映像信号用の色空間である。色差とは、RGB の各色から輝度成分の Y を差し引いた信号のことである。人間の目は明るさには敏感であるが色には敏感でないという視覚特性を利用しており、この色空間では輝度と色差を使って色を表現している。三本線を抽出するにあたり、以下の値を計算し、三本線の組み合わせを選択する (図 4(b)).

- 指根元同士の色差信号
- 指根元と線分の中心点の色差信号
- 自身ともう一つの他線とで作る中間線上の輝度信号と色差信号の標準偏差

3.4 ライン補正処理

提案手法では、指を閉じた状態の手を認識するため、指の第二関節が接触していることを前提としている。そのため、第二関節のあたりで線が途切れ、図 5(a)のように指の付け根まで線分が伸びない場合がある。線分が根元まで

届いていないとキーポイントとなる指と指の間の窪みが検出できない。キーポイントの検出に失敗すると、掌紋領域が正しく抽出できないため認証精度著しく低下する。そこで、以下のアルゴリズムにより線分の長さを補正する。

人差し指と中指の間に検出されたポイントから手のひらの方向に伸びる線分をベクトル $v_{11} = (x_0, y_0)$ 、中指と薬指の間に検出されたポイントへ伸びる線分をベクトル $v_{12} = (x_1, y_1)$ とし、それらのベクトルがなす角を θ_1 とする。また、小指と中指の間に検出されたポイントから手のひらの方向に伸びる線分をベクトル $v_{22} = (x_3, y_3)$ 、中指と薬指の間に検出されたポイントへ伸びる線分をベクトル $v_{21} = (x_4, y_4)$ とし、それらのベクトルがなす角を θ_2 とする。それぞれのベクトルの内積を求める。

$$v_{11} \cdot v_{12} = x_0x_1 + y_0y_1 \quad (1)$$

$$v_{21} \cdot v_{22} = x_3x_4 + y_3y_4 \quad (2)$$

式 (1) および (2) の値が負の値のとき、つまり、 θ_1 と θ_2 が 90 度以上であるときは、ベクトル v_{11} と v_{22} を手のひらの根元方向にスカラー倍して伸ばす。式 (1) と (2) が 0 以上になるまで、上記の処理を繰り返し行うことで図 5(b) のように検出した線分を根元まで伸ばす。

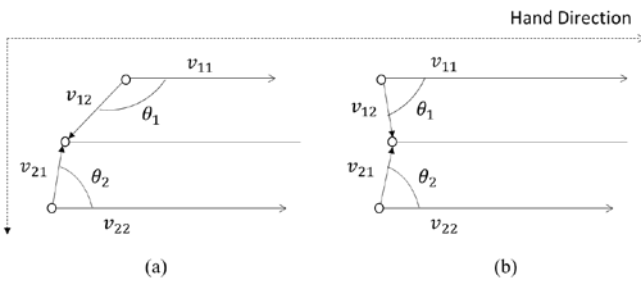


図 5. (a)第二関節で線分が途切れている場合、(b)線分の補正。

Figure 5. (a) Line is interrupted in the second joint, (b) Correction of the line segment.

3.5 掌紋領域抽出

3.4 節にて抽出した三点のキーポイントから掌紋領域（縦：160 画素，横：160 画素）を抽出する。人差し指と中指の根元を P2，中指と薬指の根元を P3，薬指と小指の根元を P4 とする。抽出方法については、以下に示す二種類の方法を検討した。

方式 1 は、P2 と P4 を結んだ線の垂直二等分線からある一定の距離をとった点を中心点とし、掌紋領域を抽出する方法である。この方式は文献[6]で紹介されている。

方式 2 は、三本線の平均角度を求め、P3 からある一定の距離をとった点を掌紋領域の中心点とする方法である(図

6)。ある一定の距離とは、P2 と P3 の距離、P2 と P4 の距離を足して 2 で割ったものをその人の平均的な指の太さとし、それを 2 倍した距離とする。固定値ではなく検出した指の太さを基準値とすることで、カメラと手のひら間の距離の変化に対して強くなる。また、方式 1 では P2 および P4 を正確に検出する必要があるのに対し、方式 2 では P3 さえ正確に検出されていれば掌紋領域を切り出せる。そのため、方式 2 は、方式 1 よりも安定していると考えられる。

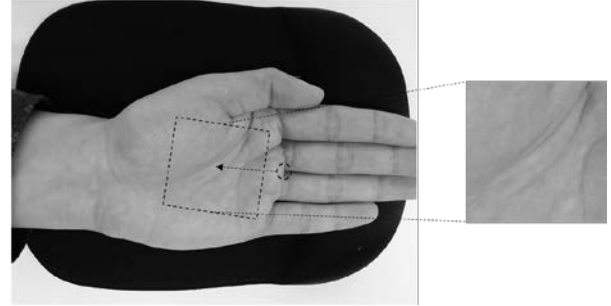


図 6. 切り出し方式 2

Figure 6. Extraction method 2.

3.6 照合処理

照合する際に登録画像と入力画像との間で掌紋領域全体および局所的に位置ずれが発生しているため、位置合わせを行ってから照合を行う。

3.6.1 位置合わせ

画像間の大域的な位置合わせには、帯域制限位相限定相関法 (Band-Limited Phase-Only Correlation: BLPOC) [4] を用いる。高精度に局所領域の平行移動量を推定するために、階層探索を利用する。階層数は 3 とし、 160×160 画素の入力画像に対して、4 画素ごとに平均値を求めることで縮小画像 80×80 画素および 40×40 画素の階層画像を生成する。ウィンドウサイズを 32 とし、一階層目である 40×40 画素の画像に対して BLPOC を用いて平行移動量を推定する。次に、二階層目の 80×80 画素の画像を 16 ブロックに分割する。そのときに、入力画像は、一階層目で求めた平行移動量を考慮した位置からブロックを抽出する。そして、BLPOC を用いて登録画像と入力画像のそれぞれのブロック間の平行移動量を求める。三階層目で二階層目と同様に 16 ブロックに対して BLPOC を用いて平行移動量を求める。図 8 左に算出した基準点(白い円)および移動量をずらした対応点(黒い円)を示す。それぞれのブロック間で計算される BLPOC 関数の最大ピークの値が閾値を超える対応点ペアを求める。本論文では、閾値を 0.3 とし、閾値を超えた基準点の座標を $\{(x_0, y_0, \dots, x_n, y_n) | 0 \leq n \leq 15\}$ 、対応点の座標を行列 $(x'_0, y'_0, \dots, x'_n, y'_n | 0 \leq n \leq 15)$ とする。ここで、座標の点数 n は必ずしも 16 にはならず登録画像および認証画像によって変動する。本論文では、推定に必要な座標の点数が 3 点以上であるアフィン変換で大域的な画像変形を近似

する(図 8 右). アフィン変換による大局的な位置補正を行うために, 行列 X および行列 Y を以下のように定義し最小二乗法にてアフィン変換用の行列 A を求める.

$$X = \begin{bmatrix} x_0 & y_0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 0 & 0 & 1 & 0 \\ 0 & 0 & x_0 & y_0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & x_n & y_n & 0 & 1 \end{bmatrix} \quad (3)$$

$$Y = \begin{bmatrix} x'_0 \\ \vdots \\ x'_n \\ y'_0 \\ \vdots \\ y'_n \end{bmatrix} \quad (4)$$

$$XA = Y \quad (5)$$

$$A = (X^t X)^{-1} X^t Y \quad (6)$$

ここで $(X^t X)^{-1}$ の演算では 6 次正方行列の逆行列を求めることになるため計算量が多く実用的でない. 以下のように式変形を行い, 3 次元の逆行列の演算とすることで高速化を行う.

$$X' = \begin{bmatrix} x_0 & y_0 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \quad (7)$$

$$Y' = \begin{bmatrix} x'_0 & y'_0 \\ \vdots & \vdots \\ x'_n & y'_n \end{bmatrix} \quad (8)$$

求めた行列 A を用いてアフィン変換を行う. デジタル画像の各画素は, x 軸と y 軸に対して, それぞれ正の整数で表される座標値を持っている. 行列 A を用いて線形変換を行うと, 変換後の座標値が正の整数とならず, 入力画像の各画素値をどの画素に移すべきかを考える必要がある. 変換後の座標値が, 正の整数とならない場合は, その周りの画素値を用いて決定する補間 (interpolation) という作業が必要となる. 補完の種類には, 最近傍法 (nearest neighbor), 双一次補完法 (bi-linear interpolation) および双三次補完法 (bi-cubic interpolation) などがある[7]. 本論文では, BLPOC にて求めた移動量および対応点の座標値を倍精度浮動小数点実数として表現しているため, 双一次補完法 [8] を用いる. 双一次補完法では, 求めたい位置の画素値を, 周り 4 点の画素値を用いた線形補完により計算する.

図 7 中の A,B,C,D の座標をそれぞれ (i, j) , $(i, j + 1)$, $(i +$

$1, j)$, $(i + 1, j + 1)$ としたときに, P の座標を (u, v) とする. 以下のステップにて, P の座標の画素値を求める.

Step 1. A および B の影響度を計算し, E とする.

$$f(i, j + v) = [f(i, j + 1) - f(i, j)]v + f(i, j) \quad (9)$$

Step 2. C および D の影響度を計算し, F とする.

$$f(i + 1, j + v) = [f(i, j + 1) - f(i, j)]v + f(i + 1, j) \quad (10)$$

Step 3. E と F の影響度を計算し, P とする.

$$f(i + u, j + v) = (1 - u)(1 - v)f(i, j) - (1 - u)vf(i, j + 1) + u(1 - v)f(i + 1, j) + uvf(i + 1, j + 1) \quad (11)$$

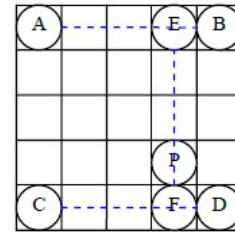


図 7. バイリニア補間アルゴリズム
Figure 7. Bilinear Interpolation Algorithm

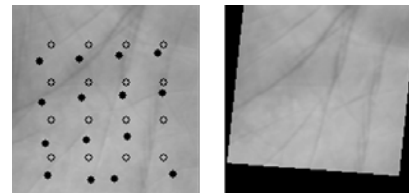


図 8. 左:登録画像の基準点(白い円)および認証画像の対応点(黒い円),右:認証画像をアフィン変換した画像
Figure 8. left: Reference point of the registered image(white circle) and Corresponding point of authentication image(black circle), right: Affine transformation image.

3.6.2 照合

画像間の大局的な位置合わせによって, 画像間の大局的な変形は補正されるが, 局所的な変形は残ったままである. 例えば, 局所的な変形は, 指の広げ方の違いなどによって手のひらがひずむことで生じる. 正確に認証するためには, 局所的な変形を考慮して照合スコアを算出する必要がある. そのために, 画像サイズを中心領域である 96×96 画素の領域をウインドウサイズ 32×32 画素の 9 ブロックに分割

表 1. 手のひらの検出結果

Table 1. The result of hand detect.

親指	切り出し方法	手のひら検出	掌紋領域検出	照合
CLOSE	方式 1	91.2% (971/1,064)	91.2% (971/1,064)	76.3% (812/1,064)
	方式 2			79.8% (850/1,064)
OPEN	方式 1	96.2% (1025/1,064)	95.6 (1,018/1,064)	86.6% (922/1,064)
	方式 2			90.6% (964/1,064)
平均速度 [μs]		8,801	2,269	16,612

表 2. 計測に使用した計算機の性能

Table2. Specifications of the computers

端末	OS	CPU	コア数	RAM
PC	OS X Yosemite	2.8GHz Intel Core i5	2	16GB

し、各局所画像ブロック間で BLPOC を計算して移動量を推定する。入力画像の掌紋領域において、推定した移動量だけずらした位置から局所画像ブロックを再度切り出す (図 9 右)。9 ブロック全ての局所画像ブロックペアに対して、BLPOC を計算する。得られた周波数データの平均をとり、その最大ピークの値を照合スコアとする。

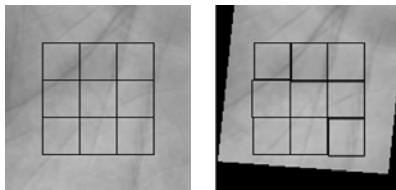


図 9. 左:登録画像の認証に使用する領域、
右:認証画像の認証に使用する領域

Figure 9. left: Area used for authentication of the registered image,

right: Area used for authentication of the authentication image.

4. 性能評価

手のひら全体が映った画像 (以降、手のひら画像と呼ぶ) に対して提案するアルゴリズムを適用した時の掌紋領域の検出率および速度を計測した。手のひら検出する際に、指を閉じた場合(CLOSE)と開いた場合(OPEN)の画像があるためそれぞれで比較を行った。手のひら画像から掌紋画像を切り出すための方法についても 2 種類実装を行ったため比較を行った。表 1 にそれぞれの結果を示す。手のひら検出率とは、手のひら画像から手のひらの目安となる指間の線分が検出された割合である。掌紋領域検出率とは、指間の線分の座標を元に正しく掌紋領域を切り出すことが出来た割合である。照合結果は予め、登録しておいた掌紋領域と照合した際に閾値 0.3 以上のスコアが得られ認

証に成功した割合である。照合結果はあくまで、手のひら画像から掌紋領域が正しく検出できていることを判別するための指標であり、一般的に言われている本人拒否率とは異なることに注意されたい。全体数は 532 名の両手データである。

親指の開閉については開いた方式の方が 5% 高くなることが分かった。この原因としては、親指を閉じた場合には親指と人差し指の間の線分も抽出する必要があり、検出するためのハードルが高くなったためだと考えられる。手のひらを検出出来なかった割合は、親指を閉じた場合で 8.8%,開いた場合で 3.8%であった。手のひら検出出来なかった画像を見ると、小指が開いていたものや指全体が開けられていない画像が含まれておりそれらが原因であると考えられる。

切り出し方式については、照合結果が 3% 程度高くなっていることから方法 2 のほうが正しく掌紋領域抽出できていることが分かる。方式 1 では 2 点のキーポイントを基準にして掌紋領域を抽出するため片方の座標がずれていると影響が大きいことが原因として考えられる。方式 2 では 1 点のキーポイントから掌紋領域を抽出するため、比較的安定して領域抽出できると考えられる。

表 2 の環境で実行したときの平均速度に関しては、照合、手のひら検出、掌紋領域抽出の順で処理時間がかかっている。手のひら検出に関しては、手のひら画像の背景によって処理時間が異なる。今回用いた実験データでは手のひら画像の背景に手のひら以外の物体が映っていないが、他の物体が映っている場合にはライン検出処理の段階で多くの線分が検出されるため処理時間が増加することを確認している。一方、掌紋領域検出および照合にかかる時間については、手のひら画像の背景に関係なく一定時間内で処理が実行できる。

5. おわりに

本論文では手のひらをかざした時にできる指間の線に着目した掌紋認証を提案した。本方式では、指間の線を正確に抽出し手のひらの角度および指の太さを計算することで、手のひらの回転および拡大・縮小に対応している。そのため、認証を行うユーザは手のひらの向きおよびカメラから

の距離を意識することなく認証することができる。本提案方式の有効性を示すため 532 名のデータセットを使用して評価を行い、サービスに適用可能なレベルの精度が得られていることが分かった。

謝辞 本論文の作成にあたり、東北大学（現アズビル（株））青山様に多大なる助言を頂いた。ここに感謝の意を表する。

参考文献

- [1] A. K. Jain et al., "An introduction to biometric recognition," IEEE Trans. Circuits and Systems for Video Technology, vol.14, no.1, pp.4-20, Feb.2004.
- [2] 太田ほか, "非接触型掌紋認証を用いたリモート認証システムの提案と評価", 情報処理学会シンポジウム論文集, vol.2010, pp.639-644, Oct.2010.
- [3] E. Yoruk et al., "Shape-based hand recognition," IEEE Trans. Image Processing, vol.15, no.7, pp.1803-1815, July 2006.
- [4] 青山ほか, "携帯電話向け非接触掌紋認証アルゴリズム," 電子情報通信学会論文誌, Vol. J96-A, No.5, pp.250-263, 2013.
- [5] 池ほか, "自然な手振りによるハンドジェスチャ ユーザインターフエース," 東芝レビュー, vol.67, no.6, pp.36-39, 2012.
- [6] David Zhang et al., "Online Palmprint Identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.25, no.9, pp.1041-1050, Sep.2003.
- [7] D. Han, "Comparison of commonly used image interpolation methods", Proc. the 2nd Int'l Conf. Computer Science and Electronics Engineering, pp.1556-1559, 2013.
- [8] S. Wang et al., "An image scaling algorithm based on bilinear interpolation with VC++," J. Techniques of Automation & Applications, pp.44-45, 2008.