

顔顔追跡による継続認証システムの構築 Construction of Continuous Authentication System by Face Tracking

田所 龍介†
Ryusuke Tadokoro

納富 一宏†
Kazuhiro Notomi

1. はじめに

現在、顔追跡技術の進歩は著しく、ビデオカメラ等から取得した動画情報から顔の位置や姿勢をリアルタイムで特定することが可能となっている^[1]。取得した情報は、その人物の顔の形状や向きなども求めることができるので、その生体情報を利用し、本人認証に応用できる可能性がある。生体情報を用いた本人認証（バイオメトリクス認証）技術の研究は盛んであるが、そのほとんどは行動・動作開始時点での一度きりの認証を前提としている。したがって、認証終了後に当該人物の交替が行われた場合、成りすましが可能となり、認証後の利用者が本人であることを保証できない。この問題を解決するためには、①継続的に認証を実行できること、②利用者が無意識的に生体情報を提示でき、③リアルタイムで認証が行えること、の3点に注目することが重要であると考えられる。

ここで、生体認証技術のうちの一つである顔認証は、非接触型であるため、認証に適した撮像アングルやタイミングの条件を明らかにすることで、利用者は無意識的に生体情報を提示することができる可能性がある。すなわち継続的な認証を実現可能であると考えられる。既存の顔認証システムの多くは導入に際し、撮像のための装置・機材や専用の処理ソフトウェアのためのコストが掛かる。さらに、特定の場所への据え置き型であるものが多いため可搬性が低い。このことから、可搬性が高く、導入も容易に行えるシステムが望ましい。

本研究ではリアルタイムで継続的に顔認証が行えるシステムを構築することを目的に、汎用的な Web カメラを用いて Web 上で継続的に顔認証を行うシステムを試作している。カメラは一般的に市販されている Web カメラや PC やスマートフォンなどの内蔵カメラで顔追跡を行えるため、導入に必要なコストや手間を削減することができる。さらに Web ブラウザ上で動作するので、環境は限定されない。また、顔認証を行う際の時間も 0.5[sec]以内で行えるので、継続的に認証を行う際のリアルタイム性を確保できる。

本稿では、汎用的な Web カメラを用いて顔認証が行えることを確認するために、顔の向きごとの認証率、認証にかかる時間を計測し、分析・評価する。これらの結果から、これからの設計について考察する。

2. 継続顔認証システム

2.1 システムの概要

本システムは、汎用的な Web カメラと Web ブラウザ上で動作するクライアント・サーバ型システムである。本システムは、スマートフォン等の内蔵カメラでも顔追跡が行えるため、特別な装置や環境に依存しない。ブラウザが画像をサーバへ送信し、サーバからブラウザへ認証結果が送

られるまでの時間は 0.5 秒以内で行われるので、リアルタイム性を確保できる。本システムの構成を図 1 に示す。

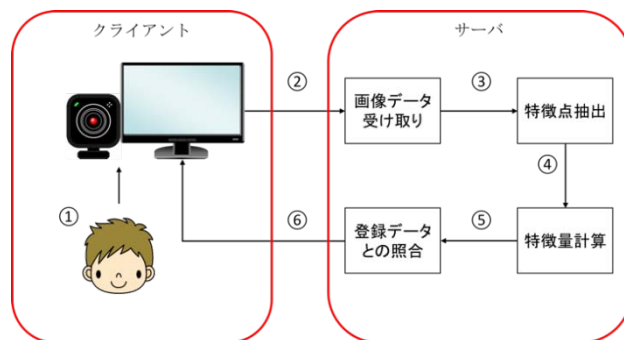


図 1 システム構成図

図 1 より、クライアント側では、①で Web カメラ等が利用者を撮影する。②では撮影した画像をサーバへ送信する。③では受け取った画像から顔を識別し、顔の特徴点を抽出する。④では抽出した特徴点から、照合に用いるためのデータを計算して求める。⑤では求めた特徴量を事前に登録してある特徴量との照合を行う。⑥では照合を行った結果をクライアント側へ送信する。

本システムはブラウザ上で動作することを前提としたため、クライアント側は HTML5 と JavaScript で構築した。サーバ側は、Python で構築している。③では画像から顔を識別し、顔の特徴点を抽出している。特徴点の抽出には、Dlib^[2]を使用する。Dlib は、Vahid Kazemi 氏らの「One Millisecond Face Alignment with an Ensemble of Regression Trees」^[3]を実装したものである。

抽出した特徴点は図 2 に示す 68 点である。

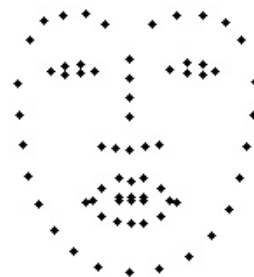


図 2 顔の特徴点（全 68 点）

2.2 照合方法

図 1 の④では、図 2 で抽出した特徴点から照合に用いるその人物の特徴量を求める。特徴量は、最初に図 3 に示す特徴点同士の距離 d を式(1)から求める。

特徴点 A を (x_1, y_1) 、特徴点 B を (x_2, y_2) とする。

† 神奈川工科大学 Kanagawa Institute of Technology

$$d = \sqrt{(m_x - P_x)^2 + (m_y - P_y)^2} \dots\dots\dots (1)$$

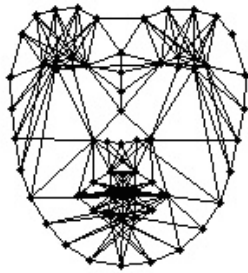


図 3 顔のモデル

図 2, 図 3 で求めた値に, 射影変換による顔向き正規化^[4]を行ったものを図 4 に示す.

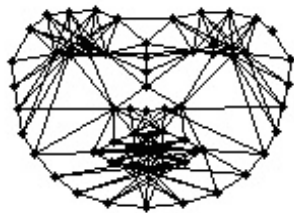


図 4 顔向き正規化

図 4 で求めた値それぞれに両目の眉間の距離で割ったものを特徴量とする.

この特徴量をあらかじめ登録しておいたデータとの相関係数を求め, データの中で一番相関係数が高かったデータを認証結果とする. また, データの登録は, 正面, 右, 左, 右上, 左上, 上, 右下, 左下, 下方向の 9 つのデータを別々のテーブルで登録しておき, 認証時に使用者の顔向きによって用いるデータテーブルを変更する.

3. 実験

3.1 実験方法

まず 20 代の男性 10 名の顔データを登録する. 登録データは, 正面, 右, 左, 右上, 左上, 上, 右下, 左下, 下方向の 9 方向を登録する. その後, 1 週間経過した後に被験者として 7 名に顔認証システムの実験を行った. データ登録・実験の際, 使用者の眼鏡の装着・未装着に関しては分けずに行った.

本システムは, PC やスマートフォン端末の内蔵カメラでも顔追跡が行えるが, 今回は Web カメラを用いて図 5 に示す形で実験を行った.



図 5 カメラの位置

被験者は 9 方向に顔を向けてもらい, それぞれの方向で 10 回撮影したときの認証率を求めた. また, その際の実行時間を計測した. 実行時間は, ブラウザがサーバへ画像を送信したときから, サーバから認証結果が送られてくるまでの時間である. 実験に用いた環境を表 1 に示す.

表 1 実験環境

名称	バージョン等
Web ブラウザ	43.0.2357.81m
Web カメラ	Logicool HD Pro Webcam C920

3.2 実験結果

20 代の男性 7 名に実験を行った際の各方向の認証率を表 2 に示す.

表 2 各方向の認証率[%]

方向	認証率	方向	認証率
正面	94.28	左	72.85
右	70.0	左下	75.71
右上	67.14	下	57.14
上	77.14	右下	60.0
左上	70.0		

1 回の認証に要する平均実行時間は 0.27[sec]であった.

4. 考察

顔認証の平均実行時間は 0.27[sec]と, リアルタイム性は十分確保できていると言える結果が得られた. また, スマートフォン端末からの実行では, 処理時間は約 0.7[sec]ほどであり, 1[sec]以内で実行できることを確認している.

認証率に関しては, 被験者が正面を向いているときが一番高い結果になった. しかし, 正面以外では最低 57.14[%]となり, 認証率が低い傾向にあった. この結果から, 特徴量を求める際に顔向き正規化を行っているが, うまく正規化が行えていない可能性が考えられる. また, 現在は, 図 4 に示す特徴点間の距離を特徴量としているが, 違う照合方法についても検討が必要である. 正面以外の撮像角度についての認証率についての性能は十分とは言えない結果になったが, 継続認証を実現する上でのリアルタイム性の確保に関しては, 提案手法の有効性が確認された.

参考文献

[1] Shaoqing Ren, Xudong Cao, Yichen Wei, Jian Sun, "Face Alignment at 3000 FPS via Regressing Local Binary Features", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1685-1692 (2014).
 [2] Dlib C++ Library, [http://blog.dlib.net/\(2015/06/10\)](http://blog.dlib.net/(2015/06/10))
 [3] Kazemi, V., Sullivan, J., "One Millisecond Face Alignment with an Ensemble of Regression Trees", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1867-1874(2014)
 [4] 山本 一真, 増田 誠, "顔画像を用いた顔向きに頑健な人物属性(年齢・性別)推定技術", OKI テクニカルレビュー第 223 号, Vol.81, No.1(2014/05)