

## 顔認証システムの構築と顔認証精度に及ぼす諸環境についての研究

### Research on the Development of Face Recognition System and Various Environments on Facial Authentication Accuracy

○長崎晃平<sup>1</sup>, 横島亮我<sup>1</sup>, 奥岡武蔵<sup>2</sup>, 金炯秀<sup>3</sup>

\*Kôhei Nagasaki<sup>1</sup>, Ryôga Yohoshima<sup>1</sup>, Musahi Okuoka<sup>2</sup>, Hyoung-Soo Kim<sup>3</sup>

Abstract: Face recognition systems are widely used, including unlocking mobile devices and managing access. In this study, we report that we are developing a face recognition system so that we can quickly confirm the identity of a large number of people. In addition, we will report on our initiatives focusing on the various environments that affect the accuracy of face recognition.

#### 1. はじめに

顔認証システムは、携帯のロック解除、入退室管理や、コンサート会場での本人確認などで幅広く使用されている。顔認証のメリットに、セキュリティ強化や利便性の両立が挙げられる。最も普及しているICカードによる認証は手軽さと低コストのメリットがある反面、なりすましなどのセキュリティの問題やICカードの紛失などによる従業員の事務作業増大による管理コストが増えるという課題もある。顔認証であれば、一度顔情報を登録することで、本人であれば手ぶらで入場が可能であり、従業員の負担も軽減できることは言うまでもない。特に、生体認証として、指紋や静脈ではなく顔を用いるメリットは次のようなものもある。食品工場で勤務する方々は衛生管理上、機械に直接接触れないように手袋をしている背景から非接触型認証である顔認証は効果的であり、様々な場面で利用が想定され、顔認証関連研究も活発に行われている。

本研究では、顔認証精度に及ぼす諸環境について着目し、諸環境下（髭、髪型、化粧、眼鏡、周囲の照度など）での計測顔データと顔認証精度についても検討する。

#### 2. 実験

##### 2.1 顔認証システム構築

顔認証システムには、ジェスチャーや音声認識によって操作ができる Kinect V2（以下、Kinect と称する）を利用した。Kinect が有する機能は、カラー画像の取得、IR 画像の取得、Depth データの取得、人の検出、人の関節の取得、音声入力・方向の取得、音声認識、顔の位置や目などのパーツの位置・表情の取得などがある。これに加えて推奨仕様を有する PC が必要である。

##### 2.2 顔認証原理

顔認証の原理には、主に 2 次元パターンマッチングと 3 次元パターンマッチングがある。2 次元パターンマッチングは、z 軸方向を使用しないことから顔が傾いたりすることで、一致率が大きく低下してしまう。3 次元パターンマッチングは、x, y, z 方向でのマッチングに加えて、拡大率の 4 つのパラメータが必要であるため計算時間と認証時間が長くなるデメリットがある。そこで Kinect が両目、鼻、唇の両端の 5 点を認識できることを利用して、その 5 点のパーツ間の長さをそれぞれ測定し、それを比較することで顔認証を行うことにした。この比較方法を用いることで、データ一つ当たりの計算回数が 10 回となり、高速化が期待できる。

##### 2.3 顔データの取得

顔データの取得には、Depth 画像と IR 画像を用いる。Depth 画像では、Kinect のセンサー面からの距離の情報を取得する。IR 画像では、顔のパーツを取得する。初めに、IR 画像から顔パーツの (x, y) 座標を取得する。次に、Depth 画像より (x, y) 座標の深さデータである z 座標を取得することにより、それぞれの顔パーツの (x, y, z) 座標を取得することができる。

顔パーツ a と顔パーツ b の距離 L は次式で求められる。

$$L = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad (1)$$

この距離を顔のパーツごとに求める。認識できる顔パーツは 両目、鼻、唇の両端の 5 個なので、計 10 個のデータが求められ、それを顔データとして登録する。

#### 3. 実験結果

顔データの認証結果について以下に述べる。2.3 節で

取得した顔データを登録してあるデータと比較し、その結果を一致率として求めた。一致率は足し算方式と掛け算方式で求めることができる。本研究では、掛け算方式<sup>[2]</sup>で一致率を算出することにした。顔データの認証では、計測データと登録データが一对多のとき、正しく認証できるかを検証した。原理をもとに作ったプログラムを用いて、認識させたい顔とダミーの顔 5 人のデータを登録し、正しく認証できるかを調べた。実際に顔が認識されている結果を図 1 に示す。顔が認識されると各パーツの座標が表示される形となっている。この結果から、一对多でも認証が成功していることが確認できた。また、登録データを 60 人分に増やしたときの認証速度も計測した。この場合は、ダミーの顔データを複製して、60 人分のデータを用意した。図示していないが、結果的に認証時間は 2ms しか増加せず、比較的、高速に認証できていることが確認できた。しかし、この認証方法では、違う顔でも認証率が高くなってしまい認証される可能性も否定できない。改善方法としては、現在の認証プログラム<sup>[1],[3],[4]</sup>で認証率が高いものを選出してから、パターンマッチングなどの精度が高い認証方式を用いることで、認証精度向上と処理時間短縮を両立させることができるのではないかと考える。今後、顔の変化（髭、髪型、化粧、眼鏡）、周囲の照度や Kinect からの距離によって一致率が変化するかどうかを報告する。

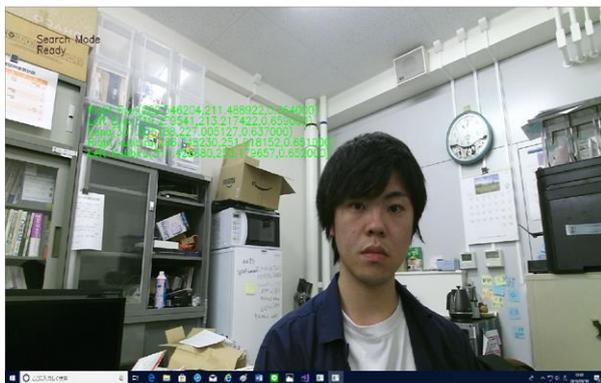


Figure 1. Face recognition result

#### 4. 参考文献

- [1] 中村薫, 杉浦司, 高田智広, 上田智章: 「KINECT for Windows SDK プログラミング Kinect for Windows v2 センサー対応版(秀和システム)」, 発行日 2015 年 5 月 25 日
- [2] 奥岡武蔵: 「卒業論文」, 日本大学理工学部電子工学科
- [3] 中村薫: 「 github KinectV2-Face-01」  
[https://github.com/K4W2-Book/K4W2Book/blob/master/C%2B%2B\(Native\)/08\\_Face/KinectV2-Face01/KinectV2/main.cpp](https://github.com/K4W2-Book/K4W2Book/blob/master/C%2B%2B(Native)/08_Face/KinectV2-Face01/KinectV2/main.cpp), 最終閲覧日 2019 年 9 月 23 日
- [4] Yoshihisa Nitta: 「Kinect V2 プログラミング NtKinect: Kinect V2 C++ Programming with OpenCV on Windows10」  
<https://nw.tsuda.ac.jp/lec/kinect2/>  
 最終閲覧日 2019 年 9 月 23 日