

VR 環境でのコミュニケーション円滑化の検討

- ジェスチャー認識機能の開発 -

Smooth communication in VR environments

- Development of gesture recognition function -

古市 真悟[†] 新徳 健[‡]
 Shingo Furuichi Takeshi Shintoku

1. 研究の背景と目的

昨今、コロナウィルスの流行によるテレワークや休校などによって VR(Virtual Reality)を用いた遠隔授業や会議のシステムに注目が集まっている^[1]。VR空間上で授業や会議を行うことによって、資料をバックに身振り手振りを用いての説明が可能となり、高い授業効果を得ることや、会議におけるコミュニケーションを円滑にすることが期待できる。しかし、VRシステムにも VRを導入するコストが高いことや、コミュニケーションが阻害される場合があるという問題点がある。そこで VRシステムで Webカメラを用いたジェスチャー入力を可能とする非接触型のインタフェースを導入することでこれらの問題点を解決できるのではないかと考えた。本研究では、ジェスチャー入力を可能とする非接触型インタフェースに必要なジェスチャー認識機能の開発を行う。

2. ジェスチャー認識機能の開発

今回作成するジェスチャー認識器では、人の骨格を認識する必要があるため姿勢推定を用いる。姿勢推定には、Googleによって2018年3月に発表された画像や動画からヒトの姿勢推定を行うモデルである PoseNet を用いる。PoseNet は画像や動画を入力すると、17個のキーポイントそれぞれの座標と確信度が得ることができ、画像や動画内の人物のポーズを推定することができる機械学習モデルである^[2]。

2.1 作成する認識器の概要

今回作成する認識器の概要を図1に示す。認識器はブラウザ上で動作し、リアルタイムで対象者が行ったジェスチャーを検出する。

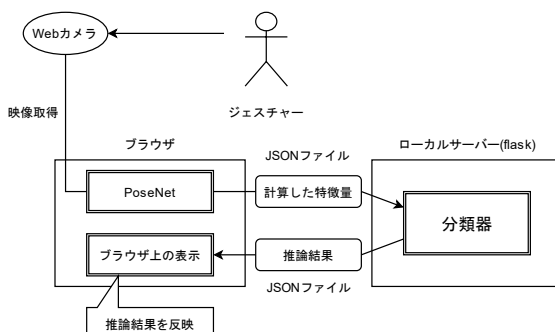


図1 認識器の概要

[†] 鹿児島工業高等専門学校専攻科電機情報システム工学専攻
 National Institute of Technology, Kagoshima College

[‡] 鹿児島工業高等専門学校情報工学科
 National Institute of Technology, Kagoshima College

作成する認識器では、ジェスチャーを特徴的なポーズが連続しているものとして図2のようにとらえる。特徴的なポーズが連続しているという一連の流れをジェスチャーとしてとらえるため、時系列を用いて学習を行う LSTM を用いて分類器を作成する。本認識器では図2の①右動作、②左動作を検出する。

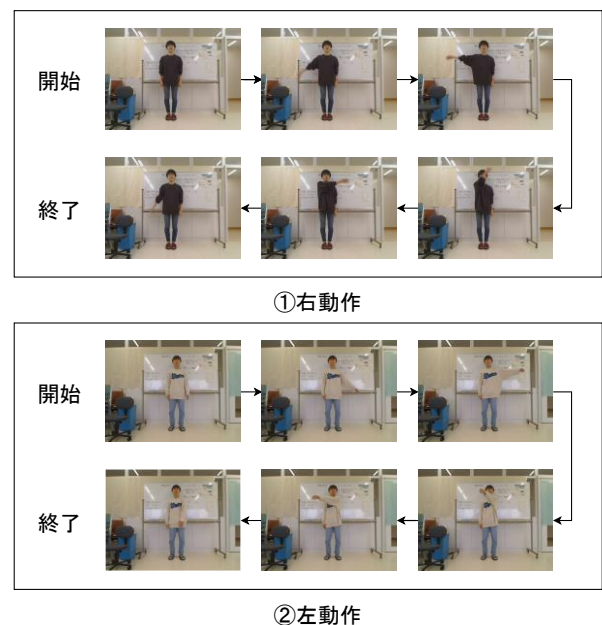


図2 検出するジェスチャー

学習に用いる特徴量として左右の手足のなす角度と各キーポイントと両肩の midpoint からの距離を用いる。

左右の手足のなす角度は、手の場合は手首-肘-肩、足の場合は足首-膝-腰の三点の座標を用いて求める。腕のなす角度を例に挙げると、肘と肩を結んだものを $\vec{a}(a_1, a_2)$ 、肘と手首を結んだものを $\vec{b}(b_1, b_2)$ とし、求める角度を θ として(1)式に示す。

$$\theta = \cos^{-1} \frac{a_1 \times b_1 + a_2 \times b_2}{\sqrt{a_1^2 + a_2^2} \sqrt{b_1^2 + b_2^2}} \quad (1)$$

PoseNet によって座標の得られる 17 のキーポイントのうち頭部、左右の肩、腰を除外した 8 個のキーポイントまでの距離を採用する。両肩の midpoint の座標を (x_1, y_1) 、距離を求めたいキーポイントの座標を (x_2, y_2) として求めたい距離を z として(2)式に示す。

$$z = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

2.2 学習に用いたデータについて

18 歳～20 歳の男女学生合計 15 人に協力してもらい右動作、左動作、上記二つの動き以外の動きを一人 20 回ずつ行ってもらいその様子を動画に撮影した。撮影した動画に PoseNet を適用し、得られた値をもとに特徴量を計算する。計算した値を csv ファイルに出力した各動作 300 個のデータ、計 900 個のデータを学習に用いる。

3. 評価

評価のための被験者は 18 歳～20 歳の男女学生合計 20 人である。

3.1 検出率と誤検出

20 人が右動作、左動作を 10 回行ったときの検出率と誤検出の回数と、平均の検出率と誤検出の回数を調査した結果を表 1 に示す。誤検出とは、ジェスチャーを行っていないのにジェスチャーが検出された場合のことや、一度しかジェスチャーを行っていないのに複数回検出するような過剰検出をまとめて指す。

表 1 各ジェスチャーの平均の検出率と誤検出の回数

| | 平均検出率(%) | 誤検出の平均回数(回) |
|-----|----------|-------------|
| 右動作 | 96 | 4.2 |
| 左動作 | 98 | 1.8 |

表 1 よりそれぞれのジェスチャーの検出率は、右動作が 96%、左動作が 98%とかなり高かったため、ジェスチャーの検出という点では問題ないといえる。しかし、どちらのジェスチャーにおいても誤検出が見られた。誤検出については、大半がジェスチャーを行っていないにもかかわらず、ジェスチャーを行っていないと判別されてしまう瞬間が生まれることによって生じる過剰検出であるが、ジェスチャーを行っていない場合の誤検出も少なからず見られた。

3.2 リアルタイム性の評価

左右の動作を終えた被験者 20 人にジェスチャーの検出がリアルタイムで行えているかどうかというアンケートを行った。結果として 20 人中 20 人がジェスチャーの検出がリアルタイムで行えているという回答が得られたため、この認識器はリアルタイムでジェスチャーの検出が行えているといえる。

4. 評価結果に対する考察

今回作成した認識器では、検出率、リアルタイム性ともに高いという結果が得られたがジェスチャーを行っていない場合に検出される誤検出とジェスチャーの過剰検出が見られたため、このまま VR システムなどに組み込むことは難しいと判断される。

過剰検出が起きるといことはジェスチャーを行っていない最中にもかかわらず、ジェスチャーを行っていないと判別されている瞬間があり、誤検出が起きるといことは、ジェスチャーを行っていないにもかかわらずジェスチャーを行っていると判別されていることがわかる。これよりジェスチャーを正確に認識することができていない場合があることが推測される。誤検出の大きな原因としては学習に用いるデータの不足とジェスチャー定義の欠陥があげられる。

今回検出するジェスチャーを図 2 のように設定したため、各動作の開始姿勢と終了姿勢が気を付けとなくっている。先ほど述べたように学習に用いるデータが少なく正確に検出することができていないことが想定されるため、これによって何も動いてない気を付けの状態でもジェスチャーとして検出されてしまうことがあったのではないかと考えられる。

学習に用いるデータ数を増やすこと以外に、ジェスチャー定義を改善することで、同じデータ数でも誤検出を減らすことが可能であるとも考察される。これを裏付ける理由として、激しく動いているときには誤検出が起こることはなく、静止している際に誤検出が起こっていたことが挙げられる。

5. おわりに

本研究は、VR システムの導入障壁の高さと、コミュニケーションが中断される場合があるという問題点を解決したコミュニケーションを円滑に行うことができる VR システムの開発に必要となる、ジェスチャー入力を可能とする非接触型インタフェースのジェスチャー認識機能の開発を行うことを目的とし、行ったジェスチャーを判別し、検出するジェスチャー認識器の開発を行った。

結論として、今回作成した認識器は検出率、リアルタイム性はともに確保されていたが、ジェスチャーを行っていないのに検出される誤検出とジェスチャーの過剰検出が確認されたためこの認識器では、VR システムに用いることはできないというのが現状である。

インタフェースで実用可能とするために以下の改善策があげられる。

- 学習に用いるデータ数を増やす。
- ジェスチャーの定義を変更する。

これらの改善策を実現することによって、誤検出を防ぐことができる考える。そして、VR システムで用いることができるインタフェースとしていくための目標として、検出可能なジェスチャーの数を増やすことがあげられる。検出可能なジェスチャーを増やすことでより拡張性を高め、様々な用途で用いることが可能なインタフェースとなる。

検出可能なジェスチャーを増やし、誤検出が起きないインタフェースとなれば、VR システムで用いることで VR システムの導入障壁の高さと、コミュニケーションが中断される場合があるという問題点を解決した VR システムの開発も可能となると考える。

参考文献

- [1] 学校教育に VR は活用できる？ デジタル教科書に続く新たな体験型学習を解説！ 株式会社 ObotAI,2022
<https://obot-ai.com/column/4395/>
- [2] tfjs-models/pose-detection at master · tensorflow/tfjs-models,2022
<https://github.com/tensorflow/tfjs-models/tree/master/pose-detection>