第 28 回日本バーチャルリアリティ学会大会論文集（2023 年 9 月）

加速度センサを用いた VR 空間を操作する全身モーション認識の研究

Research on whole body motion recognition using acceleration sensors to manipulate VR space

久保市聡 [1)]　大橋裕太郎 [2)]
So Kuboichi, Yutaro Ohashi

1)　芝浦工業大学 工学部 情報通信工学科 （〒135-8548 東京都江東区豊洲 3-7-5, af20038@shibaura-it.ac.jp）
2)　芝浦工業大学 工学部 情報通信工学科 （〒337-8570 埼玉県さいたま市見沼区深作 307, yohashi@shibaura-it.ac.jp）

概要：現在、VR での入力方法はヘッドマウントディスプレイ（HMD）と両手が中心である。本研究は物理的制約解消のため HMD ではなく加速度センサを内蔵したモバイルモーションキャプチャー mocopi を使用し、頭、両手、腰、両足の 6 点から得られる動作データを利用した、没入感を高める新システムを開発した。Long Short-Term Memory (LSTM)を活用したニューラルネットワークがセンサーデータから全身の動作パターン（立っている、蹴っている等）を判別することで、VR 空間を操作することを可能にした。

キーワード：**モーションキャプチャ, AI, 加速度センサ**, mocopi

1. はじめに

2023 年は拡張現実(AR)と仮想現実(VR)において重要な年となっている。Apple 社が高度な VR/AR デバイス「Apple Vision Pro」[1]の販売について発表した。一方 Meta 社も Meta Quest 3[2]の発売を発表するなど、世界の企業が XR デバイスの普及に動いている。

Sony 社からは、2023 年の 1 月に加速度センサを用いたモーションキャプチャ機器である mocopi[3]が発売された。これにより、全身の動きを VR 空間に映し出すことがより安価で可能になっている。

これらの動向を踏まえて、モーションキャプチャを用いた動きの分析とその結果を VR 空間に適応する手法を探求することにより、現在の VR 操作にさらなる没入感と自然さが与えられると考える。本研究では、その手法の開発と可能性を探ることを目指す。

2. 関連研究・技術

2.1 モーション判別に関する研究

市川ら[4]は、Kinect for Windows V2[5]を用いて、カメラの前に立つユーザの骨格情報をもとにジェスチャを認識して VR 空間を操作する研究を行った。その結果、215 個の動作では学習データが足りず、認識精度が 70%から 80%であり、低いという結果が出た。また、Recurrent Neural Network (RNN)を使用した学習の際に、過学習が起きたという考察がなされた。

2.2 Kinect V2

Kinect V2 は RGB カメラや深度センサ等のセンサを用いることで骨格や表情、音声を認識することができるものである。今回はカメラの画角に全身がいなければならないという物理的制約をなくすために、使用しない。

2.3 mocopi

mocopi は 2023 年に Sony 社が発売した 6 個の加速度センサを用いて全身の動きを取得できるモーションキャプチャ機器である。スマートフォンと mocopi が Bluetooth 接続をすることで、ユーザの全身モーションをスマートフォン上のキャラクターが再現する。さらに、ゲーム開発プラットフォームである Unity[6]と UDP 通信をすることで、開発者がリアルタイムでモーションデータを使用できる。

3. 研究内容

3.1 本研究の概要

本研究の目的は、加速度センサである mocopi を用いて動きを判別して VR 空間を操作することである。本研究で開発するモーション判別システム、および VR 空間への応用の構成を図 1 に示す。

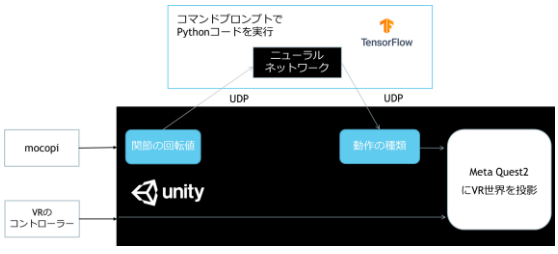

図 1 モーション識別システム及び VR 空間操作の構成図

mocopi からの情報をもとに Unity の環境内で 3D アバターが現実のユーザの動きを再現する。その 3D アバターの関節の回転データをもとにジェスチャを認識し、VR 空間に影響を与える。Python 環境で 3D アバターの関節の回転データを利用するのに Tensorflow[7]ライブラリを利用している。TensorFlow は 2015 年に Google によって開発された機械学習ソフトウエアライブラリである。使用することによって、ニューラルネットワークを構築できる。Python 環境で作成した識別器を用いてジェスチャを判定し、その結果を Unity 環境に送り込み、VR 空間を操作する。VR 空間上でモーションに合わせてイベントが起こるギミックを複数用意した。ユーザの装着する HMD にその空間が表示される。

ここで重要視したことが 2 点ある。1 点目はカメラを用いないことで HMD を着用して周りの世界が見えない状況でも、「ジャンプ」のような大きい動きが許されることである。もう 1 点は独自に様々な人の特定の動きをモーションデータとして取得し、より多くの人に適応するようなデータセットを作成することである。

3.2 ジェスチャの認識について

今回は、モーションの識別に機械学習を用いた。その中でも、モーションデータとモーションの種類を結び付けるために、教師あり学習を選択した。

モーションデータは、87×30 の 2610 の浮動小数点数によって表現される。87 は、一度に Unity 上で動きを再現する 3D アバターの関節の回転や位置を表す数字の総数である。30 は、記憶するモーションデータのフレーム数を示しており、0.1 秒ごとに 3 秒間のデータを記録するために必要となる。

3.2.1 ジェスチャの取得方法について

様々な人からモーションデータを取得するために、モーション取得ワールドを作成した。具体的な様子を図 2 に示す。カウントダウンが 5 から始まり、0 になるタイミングでユーザは「蹴る」「殴る」のような動きをする。すると自動で 3 秒前から 0 秒まで、0.1 秒おきに各関節の回転値や位置が保存される。今回は「蹴る」「殴る」「投げる」「静止する」「ジャンプする」「鳥のように滑空する」「箱を開ける」の 7 種類の動きを取得した。

図 2 モーション取得ワールドの様子

3.2.2 RNN によるジェスチャ認識について

過去 3 秒間の動きの中に、7 種類の動きを識別するのに十分なデータがあると仮定した。VR 体験中、0.1 秒ごとに、過去 3 秒間のモーションデータを取得する。具体的には、各時間ステップで 87×30 のデータを取得する。これらのデータは TensorFlow を使用したニューラルネットワークに送信され、モーションの判別が行われる。

データセットの作成には 3.2.1 で説明したモーション取得ワールドを使用した。18 人の被験者に 7 種類の動作を複数回行ってもらい、合計で 376 のモーションデータを集めた。

ニューラルネットワークには、時系列データを扱える RNN を使用する。ニューラルネットワークの構築には Keras と TensorFlow を python 上で利用している。RNN のレイヤーとして時長期的な依存関係の学習を行うことができる、LSTM を用いている。図 3 に本研究で用いた Keras によるニューラルネットワークの構成を示す。活性化関数として softmax 関数を使用している。

4. VR 空間への適応

4.1 VR 利用時のシステム

「蹴る」「殴る」「投げる」「静止する」「ジャンプする」「鳥のように滑空する」「箱を開ける」の 7 つのモーションを、作成したシステムで識別する。識別されたモーションの種類とプレイヤーの位置を参照することで、VR 空間を操作した。VR 空間の構築にはユニティ・テクノロジーズの開発したゲームエンジン Unity を使い、出力には Oculus Quest2 を使用した。

4.2 蹴る

図 4 は「蹴る」モーションの例である。ユーザが VR 空間の中で石の前に立ち、「蹴る」動きをすると、石が的に目掛けて飛んでいく。

4.3 殴る

図 5 は「殴る」モーションの例である。ユーザが VR 空間の前で大きな岩の前に立ち、「殴る」動きをすると、岩が消え、破片が飛ぶエフェクトが表示される。

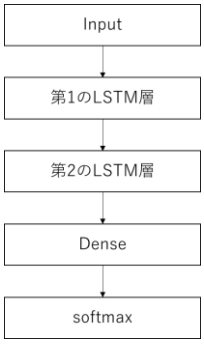

図 3 ニューラルネットワークの構成

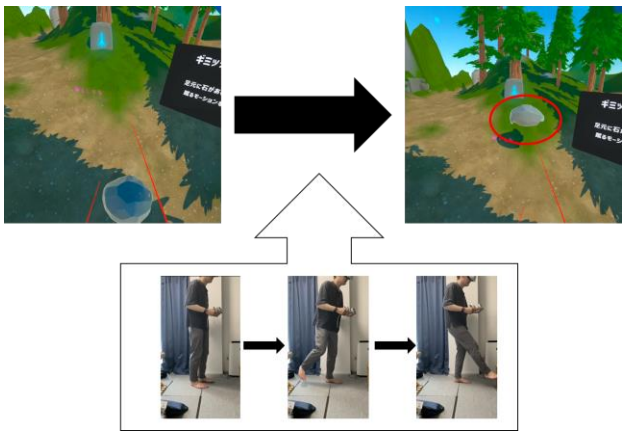

図 4「蹴る」モーションの例と VR 世界への影響

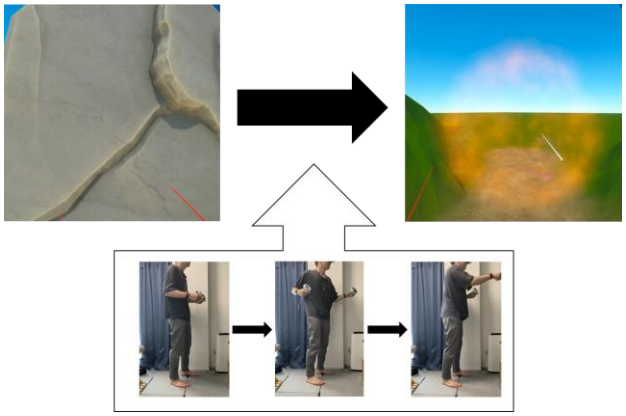

図 5「殴る」モーションの例と VR 世界への影響

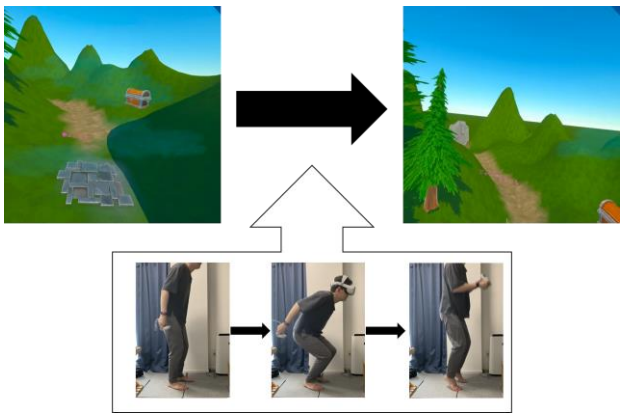

図 6「ジャンプ」モーションの例と VR 世界への影響

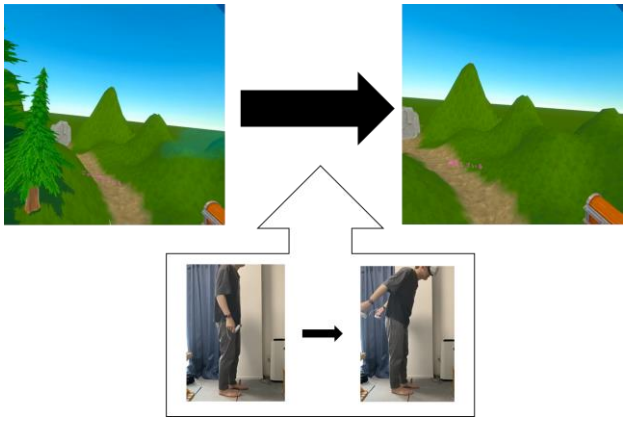

図 7「滑空」モーションの例と VR 世界への影響

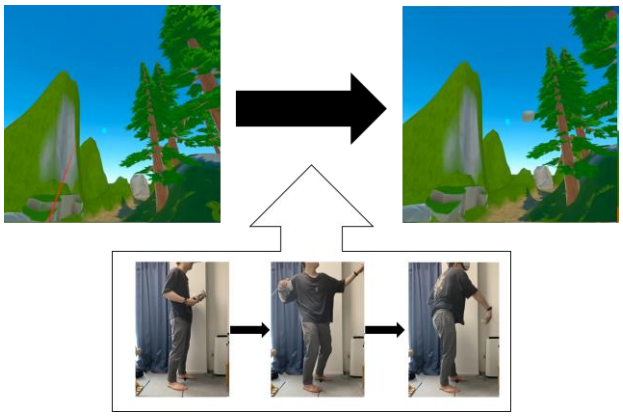

図 8「投げる」モーションの例と VR 世界への影響

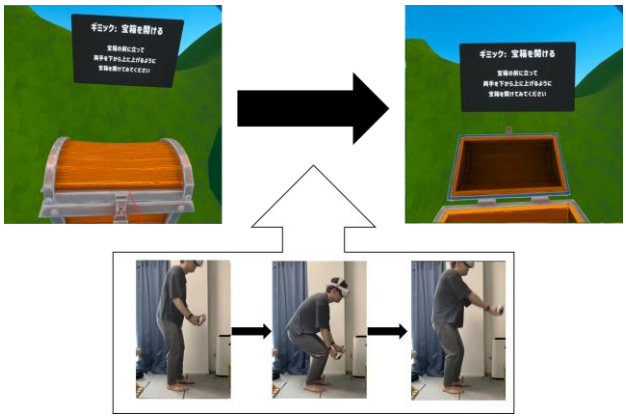

図 9「箱を開ける」モーションの例と VR 世界への影響

4.4 ジャンプする

図 6 は「ジャンプ」のジェスチャの例である。ユーザが上向きに風が出ているところに立ち、「ジャンプ」の動きをすると空中に移動し、とどまり続ける。

4.5 滑空する

図 7 は「滑空する」のジェスチャの例である。ユーザがジャンプをして空中にいる状態で滑空をすると自分の目線の方向に降下をする。

4.6 投げる

図 8 は「投げる」のジェスチャの例である。ユーザが目線を空中の風船に合わせて「投げる」ジェスチャを行うと、石が手元から風船に向かって飛んでいく。

4.7 箱を開ける

図 9 は「箱を開ける」のジェスチャの例である。ユーザが箱の前に立ち、両手を下から上に動かすように「箱を開ける」ジェスチャを行うと、目の前の箱が空く。

5. 評価実験

今回は集めたデータセットのうち 8 割にあたる 301 データをトレーニングデータ、2 割にあたる 75 データをバリデーションデータとして分類した。75 データを 7 つのモーションに予測し、認識率を調べた。

表 1 実験結果

	Precision	Recall	F1-score
蹴る	0.69	0.82	0.75
殴る	0.69	0.85	0.76
投げる	0.86	0.75	0.8
静止する	0.81	0.95	0.88
ジャンプ	1	0.43	0.6
滑空	1	0.62	0.77
箱を開ける	0.6	0.5	0.55

表 1 に実験の結果を示す。ここで Precision は正しくモーションが正と予測された割合である。Recall はモデルがどれだけ見つけられたのかを表す割合である。正しく正と出た数を、誤って偽と判断されたものと正しく正と判断されたものの合計で割ったものである。F1-score は Precision と Recall の調和平均である。全体的な認識率は 77%であった。

6. 考察

先行研究では認識率が 70%から 80%であり、低いという評価であったが、今回の実験は 77%という結果であった。このモーションの中で「箱を開ける」モーションの F1-score が最も低かった。箱を開けるという言葉で想像できる動きが分かりづらいことが起因していると考えられる。今回のモーション取得ワールドでは、箱を開けるというモーションの際に目の前に箱を用意して、イメージを湧きやすくする工夫もしたが、それでも精度が低いため、日常的に思いつきやすい動きにすることが精度向上に貢献すると考える。

今回の実験では精度に絞って調査した。今後はこのシステムを生かしていかに没入感が高い VR 体験に貢献できるのか、という部分についても研究していきたい。

謝辞

株式会社 Flamers の皆様、芝浦工業大学の皆様にモーションデータ収集の多大なご協力を頂きました。ここに感謝の意を表します。

参考文献

[1] Apple, Inc: Apple Vision Pro, https://www.apple.com/apple-vision-pro/ (2023.07)

[2] Meta Platforms,inc: Meta Quest 3, https://www.meta.com/jp/quest/quest-3/ (2023.07)

[3] Sony Corporation: mocopi, https://www.sony.jp/mocopi/ (2023.07)

[4] 市川ひまわり，新田善久：RNN を用いた VR 空間を操作する為のジェスチャ認識の研究，研究報告ヒューマンコンピュータインタラクション（HCI），2019-HCI-183，2，1-7，2019.

[5] Microsoft: Kinect for Windows, https://learn.microsoft.com/ja-jp/windows/apps/design/devices/kinect-for-windows/ (2023.07)

[6] Unity Technologies: Unity, https://unity3d.com/jp (2023.07)

[7] TensorFlow: TensorFlow, https:///www.tensorflow.org/ (2023.07)