

OpenPose と MocapNET を用いた 360 度 VR 動画からモーションを生成するウェブサービス

岡田 義広, 永山 寛二, 木佐貫 浩司

九州大学 データ駆動イノベーション推進本部 次世代型オープンエデュケーション推進部門

okada.yoshihiro.520@m.kyushu-u.ac.jp

Web Service for Motion Generation from 360-degree VR Videos Using OpenPose and MocapNET

Yoshihiro Okada, Kanji Nagayama, Hiroshi Kisanuki

Division of Next generation Open Education promotion, Data-Driven Innovation Initiative, Kyushu University

概要

本稿では、360 度 VR 動画の人物動作解析を目的とした OpenPose と MocapNET を用いた動画からモーションを生成するウェブサービスについて述べる。著者らは既に 360 度 VR 画像/動画の VR ツアーのためのフレームワークを提案している。演習/実験の過程や手順を事前に理解するのに役立つため 360 度 VR 画像/動画の VR ツアーは、対象となる演習/実験科目の教材として有用である。また、著者らは多地点 360 度 VR 動画のためのフレームワークも提案している。これらフレームワークを用いた 360 度 VR 動画による VR ツアーにおいて、動画内の人物動作を解析し学生にコメントを与える等の支援ができれば、教育効果が高まると考えられる。そこで著者らは、人物動作解析の初期段階として、OpenPose と MocapNET を用いた動画からモーションを生成するウェブサービスを開発した。OpenPose と MocapNET は、良く知られた人の 2D 姿勢推定とモーション生成の AI ツールである。これら生成 AI ツールを使うには、高性能 GPU を含む高性能コンピュータを準備し、多くの依存ライブラリを含む多くのプログラムコードをインストールする必要があり、教員のような専門知識のないユーザにとって大きな負担となる。そこで著者らは、OpenPose と MocapNET を用いた動画からモーションを生成する機能をウェブサービスとして実装した。本稿では、OpenPose と MocapNET がウェブサーバ上でどのように連携し動作するかを説明し、いくつかのモーション生成例とそれらのパフォーマンス結果を示す。

1 はじめに

九州大学は、2022 年 4 月に学内の DX を推進するため、データ駆動イノベーション推進本部を設立した。複数の部門から構成されており、その一つが我々の所属する次世代オープンエデュケーション推進部門 (NOE 部門) である。NOE 部門のミッションは、教育 DX を推進することである。特に、XR : eXtended Reality (VR : 仮想現実、AR : 拡張現実、MR : 複合現実) 等の最新 ICT を活用した教材提供や、大学教員の教材作成支援等である。実験室や演習室の VR ツアーは、学生が実験や演習を行う前に、室内の機器や操作方法を理解するのに役立つ。このような VR ツアーの作成も NOE 部門の活動の一つである。

我々は既に、ウェブベースの VR ツアーを作成するためのフレームワークとそれらを活用して制作された 360 度 VR 動画のナビゲーション VR ツアーと 360 度 VR 画像および PCD: 点群データを使用したウォークスルー VR ツアーの提供を行っ

ている[1-5]。例えば、九州大学中央図書館の VR ツアーを作成した。昨年、我々は VR ゴーグルに対応するウェブベースの多地点 360 度 VR 動画の新しい開発フレームワークを提案した[6]。自由視点動画や自由視点ライブストリーミング動画のサービスは数多く存在するが、多地点 360 度 VR 動画やそのようなサービスはほとんどない。多地点 360 度 VR 動画を撮影するには複数の 360 度 VR カメラが必要であるが、そのような動画は通常の 360 度 VR 動画や自由視点動画よりも没入感がより高い。そのため、VR ゴーグルに対応する多地点 360 度 VR 動画は教育用途に有用であると言える。

360 度 VR 動画中の人物動作を解析して演習中の学生にコメントを与える等の支援ができれば、教育効果が高まると考えられる。そこで我々は、人物動作解析の初期段階として、OpenPose [7] と MocapNET [8] を用いた動画からモーションを生成するウェブサービスを開発した。OpenPose と MocapNET は、良く知られた 2D 姿勢推定とモーション生成の AI ツールである。しかし、これらを使用するには、高性能 GPU を含む高性能コンピ

ユーザを用意し、多くの依存ライブラリを含む多くのプログラムコードをインストールする必要がある。これは、大学教員のような専門知識のない初心者ユーザには負担が大きい。そこで我々は、通常の動画や人物動作の 360 度 VR 動画からモーションデータを生成する、OpenPose と MocapNET を利用したウェブサービスを構築した。本稿では、OpenPose と MocapNET がウェブサーバ上でどのように連携して動作するのかを説明し、生成されたモーションデータ例をいくつか示すとともに、それらの精度について議論する。

本稿の残りの部分は以下のように構成されている。次の第 2 節では関連研究について説明する。第 3 節では、既存フレームワークと VR ツアー例を紹介する。第 4 節では、OpenPose と MocapNET を用いたウェブサービスを実現する方法について説明する。第 5 節では、生成されたモーションデータ例を示し、問題点等について考察する。最後に、第 6 節で本稿のまとめを述べる。

2 関連研究

LiDAR スキャナや 360 度 VR カメラなどのセンシングデバイスをリリースしている企業は数多くあります。その中のいくつかは、ユーザが 360 度 VR 画像/動画と PCD のインタラクティブなウェブベースの VR アプリケーションを作成できる商用サービスを提供している。RICOH のサービス (<https://www.theta360.biz/>) は 360 度 VR 画像をサポートしているが、360 度 VR 動画と PCD はサポートしていない。この RICOH サービスの特徴は、我々のフレームワーク[1-5]にはないウェブ編集機能を提供している点である。Matterport のサービス (<https://matterport.com/>) は 360 度 VR 画像と PCD をサポートしているが、360 度 VR 動画はサポートしていない。この Matterport サービスの特徴は、Matterport キャプチャアプリケーションと呼ばれる AI 技術に基づく強力なツールの提供です。このツールは、360 度 VR 画像と PCD から位置マップを自動生成する。我々のフレームワーク[3-5]は、OpenVSLAM[9]を使用して移動経路と位置情報を生成することができる。本稿では、360 度 VR 動画からの人物動作生成に焦点を当てている。動画からの人物動作生成に関しては、OpenPose[10]や MocapNET[11]の他にも、GitHub には video2bvh[12]、video_to_bvh[13]、video to 3DPose、bvh[14]といったツールがある。本稿では、研究の初期段階として OpenPose と MocapNET を使用したが、これら以外のツールの可用性を確認する予定である。

3 既存フレームワーク

本節では、360 度 VR 画像/動画をウェブベース VR ツアーとして提供するフレームワークを紹介する。このフレームワークは、360 度 VR 画像を用いるウォークスルー VR ツアー用と、360 度 VR

動画を用いるナビゲーション VR ツアー用の 2 つのシステムで構成されている。また、多地点 360 度 VR 動画のウェブベース VR ツアー用フレームワークも紹介する。

3.1 360 度 VR 画像/動画 VR ツアーのフレームワーク

図 1 は、360 度 VR 画像を用いたウォークスルー VR ツアー用システムと 360 度 VR 動画を用いたナビゲーション VR ツアー用システムのファイル構成を示している。

walk_map.html, video_map.html
JS { walk_map.js, walk_map_controls.js, video_map.js, video_map_controls.js }
JSM { Several JS files from Three.js library }
Assets
Images { 360VR images, optional images }
Videos { 360VR videos }
Models { optional 3D model files }
Movies { optional movie files }
Sounds { optional sound files }

図 1. 2 つのシステムの全ファイル構成

各システムには、1 つの HTML ファイル (walk_map.html または video_map.html)、1 つの主要 JavaScript ファイル (walk_map.js または video_map.js)、および 1 つの補助 JavaScript ファイル (walk_map_controls.js または video_map_controls.js) が含まれている。walk_map.js と video_map.js は、Three.js ライブラリから派生したいくつかの関数を定義している。Three.js は、ウェブ上の 3D グラフィックス関数群 WebGL を使い易くする強力なライブラリである。新しい VR ツアーを作成するとき、これら walk_map.js と video_map.js を変更する必要はない。必要なメディアファイル (つまり、Assets ディレクトリに保存される 360 度 VR 画像/動画ファイル) を準備し、walk_map_controls.js または video_map_controls.js にファイル名等を設定するだけである。ファイル名が登録されたメディアファイルは、walk_map.js と video_map.js によって適切に読み込まれる。

1) 360 度 VR 画像ウォークスルー VR ツアー

図 2 は、CT 実習室のウォークスルー VR ツアーの画面である。上図と下図は、2 つの異なる場所の画面である。マウスデバイスをメイン画面上でドラッグすることで、ユーザは視線を上下または左右に動かすことができる。各図の左側には、CT 実習室の地図が表示されている。地図上の赤いマーカーを見ることで、ユーザは自分の現在位置と

方向を把握できる。

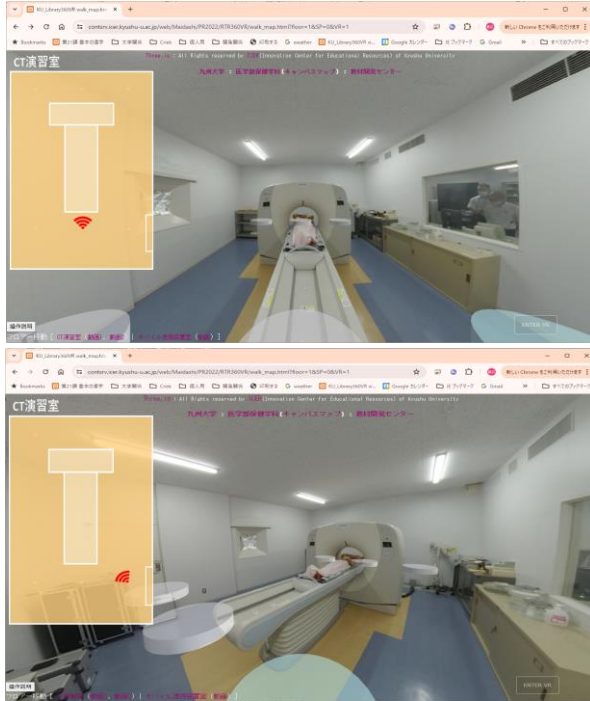


図 2. CT 実習室ウォークスルーVR ツアー

地図上には、360 度 VR 画像の撮影場所を示す小さなオレンジ色の点が表示されている。ユーザは、地図上の対応するオレンジ色の点をクリックすることで、現在の 360 度 VR 画像を、行きたい場所で撮影された画像に切り替えることができる。また、メイン画面には、360 度 VR 画像の撮影場所を示す灰色の円柱が表示されている。円柱のいずれかをクリックすることでも、現在の 360 度 VR 画像をその場所で撮影された画像に切り替えることができる。このように、ユーザは対話的に自分の位置を変更できる。

2) 360 度 VR 動画ナビゲーション VR ツアー

図 3 は、CT 実習室のナビゲーション VR ツアーの画面である。上図と下図は、異なる視線方向の画面である。メイン画面上でマウスデバイスをドラッグすることにより、ユーザは視線方向を垂直または水平に変更できる。この VR ツアーは、地図上の赤いマーカーで示された場所で撮影された 1 つの 360 度 VR 動画のみで構成されているため、地図上にオレンジ色の点は表示されていない。各画面の中央下部には、再生、一時停止、巻き戻し、早送りなど、ホームビデオを制御するのと同様に、360 度 VR 動画を制御するためのコントロールパネルがある。



図 3. CT 実習室ナビゲーション VR ツアー

3.2 多地点 360 度 VR 動画 VR ツアーのフレームワーク

この節では、ウェブベース多地点 360 度 VR 動画 VR ツアー制作用フレームワークを説明する。

```
walk_video_map.html
JS { walk_video_map.js,
      walk_video_map_controls.js }
JSM { Several JS files from Three.js library }

Assets
Images { optional images }
Videos { 360VR videos }
Transcripts { vtt files }
Models { optional 3D model files }
Movies { optional movie files }
Sounds { optional sound files }
```

図 4. 多地点 360 度 VR 動画 VR ツアー用フレームワークの全ファイル構成

フレームワークのすべての JavaScript ファイルと HTML ファイルを図 4 に示す。フレームワークには、1 つの HTML ファイル (walk_video_map.html)、1 つのメイン JavaScript ファイル (walk_video_map.js)、1 つの補助 JavaScript ファイル (walk_video_map_controls.js) が含まれている。これらの JavaScript ファイルと HTML ファイルは、前節で説明したフレームワークのファイルを統合して作成したものである。厳密に言えば、

walk_video_map.html は walk_map.html と video_map.html を統合して作成され、walk_video_map.js は walk_map.js と video_map.js を統合して作成され、walk_video_map_controls.js は walk_map_controls.js と video_map_controls.js を統合して作成された。

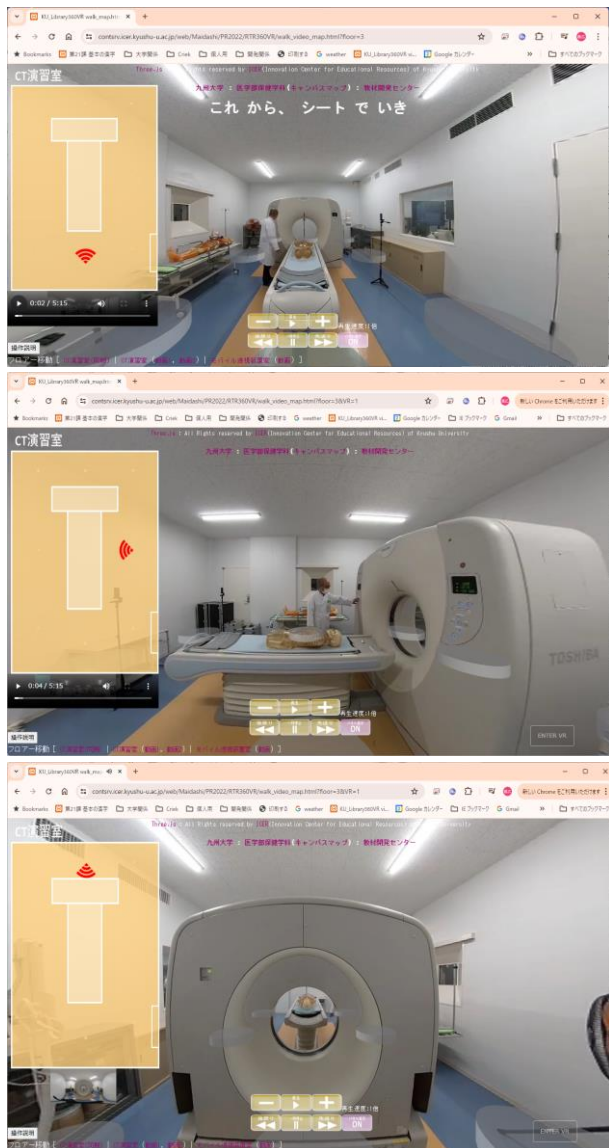


図 5. 多地点 360 度 VR 動画による CT 実習室 VR ツアー

図 5 は、多地点 360 度 VR 動画による CT 実習室の VR ツアーである。3つの画面は、それぞれ異なる場所で撮影された動画表示である。これらの画面は、図 2 と図 3 を統合したもののように見える。部屋には 5 台の 360 度 VR カメラがある。メイン画面には 5 つの灰色の円柱が表示されている。それらの位置は、5 台の 360 度 VR カメラの位置と同じである。また、各図の左側に示されている地図には、5 つのオレンジ色の点が表示されている。これらは、5 台の 360 度 VR カメラの位置である。図 2 と同様に、地図上の赤いマーカーを見ると、ユーザの現在位置と方向を把握できる。ユーザは、

地図上の対応するオレンジ色の点をクリックすると、現在の 360 度 VR 動画から、クリックした場所で撮影された動画に変更できる。メイン画面では、灰色の円柱をクリックすると、現在の 360 度 VR 動画から、クリックした場所で撮影された動画に変更できる。これにより、ユーザは対話的に自分の位置を変更できる。図 3 と同様に、メイン画面の下部中央には、再生、一時停止、巻き戻し、早送りなど、360 度 VR 動画を操作するためのコントロールパネルが表示されている。

4 動画からモーションデータを生成するウェブサービス

この節では、OpenPose と MocapNET を使用した、360 度 VR 動画から人のモーションファイルを生成するウェブサービスを紹介します。このサービスは 2 つの部分で構成されており、1 つは 360 度 VR 動画から 2D ポーズ情報である json ファイルを生成する OpenPose を使用する部分、もう 1 つは OpenPose が出力した json ファイルからモーションファイル (bvh ファイル) を生成する MocapNET を使用する部分である。以下では、まず OpenPose と MocapNET を紹介する。

4.1 OpenPose と MocapNet

図 6 に示すように、OpenPose は、複数人物の骨格(2D ポーズ)を実時間出力する非常に有名なシステムである。実行コードは GitHub[10]からダウンロードできる。

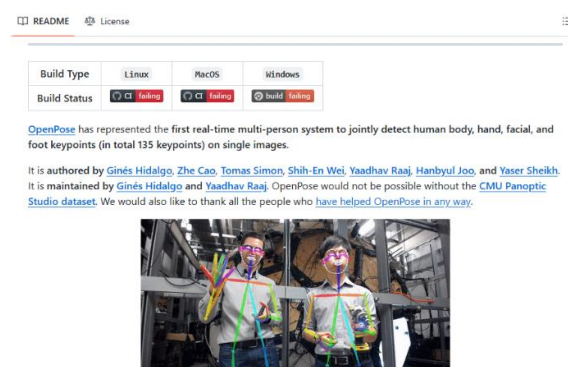
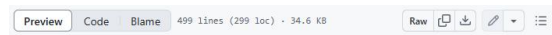


図 6. GitHub の OpenPose ページ [10].

OpenPose は Linux、MacOS、Windows で実行できる。我々のウェブサーバの OS は Windows Server 2022 のため、Windows 実行プログラム (openposedemo.exe) を使用する。実行プログラムには、CPU 版と GPU 版の 2 種類がある。CPU 版は処理速度が遅過ぎるため不適である。ウェブサーバには、GPU カードが 3 枚搭載されているため GPU 版を使用する。最新バージョンは 1.7 であるが、事前に実行速度を調べた結果、Windows 実行プログラムでは 1.6 の方が 1.7 よりも高速であるため、1.6 を使用する。付録のコマンド 1) は、Windows

実行プログラムのコマンド例である。この場合、OpenPose ディレクトリで、'bin\Openposedemo.exe' プログラムが実行される。入力用動画ファイルは'xxx.avi'であり、出力動画ファイルは'result.avi'である。出力 json ファイルは 'output' ディレクトリに保存される。OpenPose の説明書によると、Windows 実行プログラムは AVI ビデオ形式のみをサポートしている。そのため、ビデオファイル形式 'mp4' を AVI 形式 'avi' に変換し、その変換のために'ffmpeg.exe' を使用する。



MocapNET Project

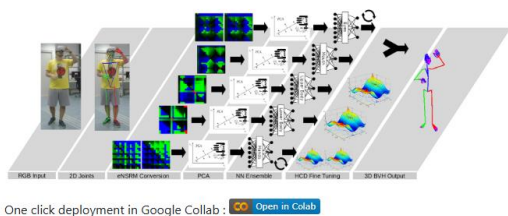


図 7. GitHub の MocapNET ページ[11].

図 7 に示すように、MocapNET は動画から bvh ファイルを出力する有名なツールであり、その実行コードは GitHub[11] からダウンロードできる。bvh ファイルは、Bio Vision Corp. によって提案された 1 人分のモーションデータファイルである。MocapNET は、人物 1 人の動画から OpenPose によって出力された人物 1 人の json ファイルから人物 1 人の bvh ファイルを生成する。

Windows 実行プログラムがないため、WSL (Windows Subsystem for Linux) 上の Linux 版 MocapNET を使用する。付録のコマンド 2)は、OpenPose が出力した json ファイルから bvh ファイルを生成する。

4.2 OpenPose と MocapNet のウェブサービス

図 8 は、OpenPose と MocapNET のウェブサービスを示している。人物 1 人の動画から bvh ファイルを生成するための実行フローである。このために、upload_video2bvh.html と upload_video2bvh.php という 2 つのシンプルなファイルを用意した。

ユーザがブラウザで upload_video2bvh.html にアクセスすると (図の(1))、図 9 に示すようなウェブページが表示される。OpenPose の入力として用いる動画ファイルを指定するためのファイル選択ボックスがある。送信ボタンを押すと、動画ファイルがウェブサーバにアップロードされ、同時に upload_video2bvh.php が呼び出される (図の(2)と(2'))。

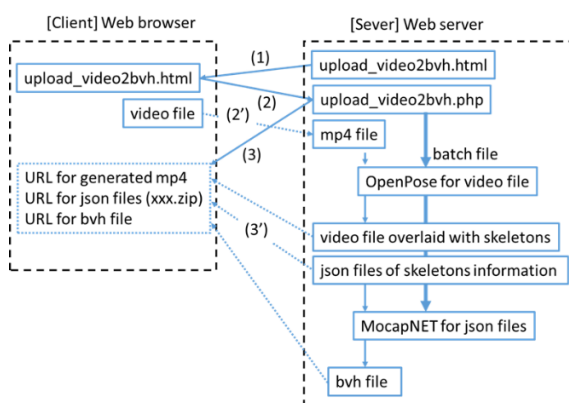


図 8. ウェブサービスの実行フロー.

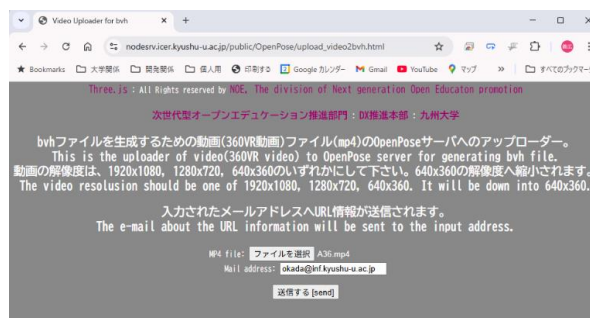


図 9. 動画アップローダーの画面.

この PHP プログラムは Windows バッチファイル (xxx.bat) を生成し、その中で OpenPose と MocapNET を実行する。バッチファイルの内容は、付録のコマンド 3) の通りである。

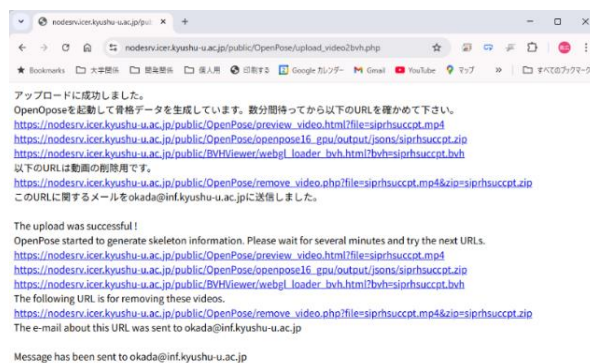


図 10. 動画アップロード後の応答画面

バッチファイルの実行後、OpenPose によって骨格 (2D ポーズ) が重ね合わされた新しい動画ファイルと、骨格情報を含む json ファイルが生成される。さらに、MocapNET によって bvh ファイルが生成される。図 10 に示されるように、これらのファイルをダウンロードするためウェブブラウザに 3 つの URL (3') が表示される。ユーザが最初の URL をクリックすると、図 11 に示すように、2 つの動画フレームが表示される。上部は元の入力動画で、下部は骨格(2D ポーズ) が重畳表示される新しく生成された動画である。ユーザが 2 番目の URL をクリックすると、OpenPose によ

って出力された json ファイルの zip ファイルがダウンロードされる。さらに、ユーザが 3 番目の URL をクリックすると、図 12 に示すように、bvh ファイルのウェブビューアーが表示される。このウェブビューアーは Three.js の例の 1 つとほぼ同じであるが、bvh ファイルのダウンロードボタンが追加されている。

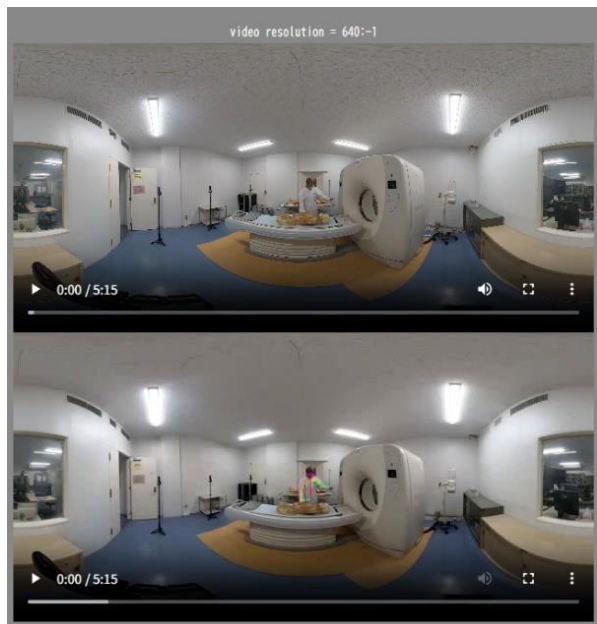


図 11. 元動画と OpenPose 出力動画

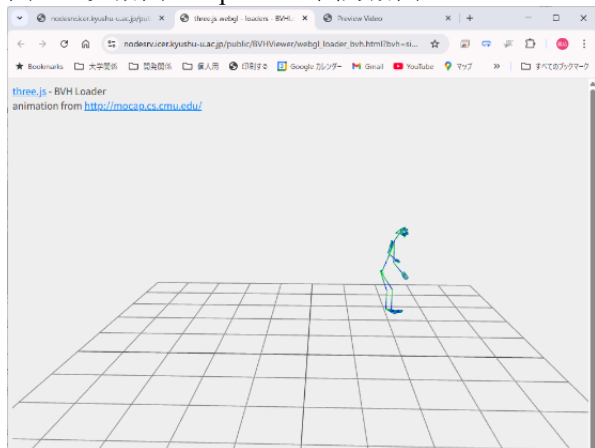


図 12. MocapNET 出力モーション

この例の場合の実行性能は次の通りである。PC の仕様は、CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz、GPU: GeForce RTX 2080 Ti × 3、MEM: 128GB であり、図 11 の 360 度 VR 動画 (1920x1080、30 秒、29.97fps) を OpenPose と MocapNET で実行した結果、それぞれ合計 24.3 秒と 27.5 秒であった。

5 動画生成例と議論

この節では、骨格 (2D ポーズ) を重ね合わせた別の生成動画を示す。図 13 は、OpenPose によって出力された動画のあるフレームを示している。

元の動画には 2 人の人物が登場するため、2 つの骨格 (2D ポーズ) が重ね合わせられている。この場合の性能結果は次の通りである。PC の仕様は先と同じである。図 13、14 の 360 度 VR 動画 (1920x960、30 秒、29.97fps) に対する OpenPose と McapNET の実行時間は、それぞれ合計 23.7 秒と 20.0 秒であった。このように、OpenPose は複数人物の骨格 (2D ポーズ) とそれらの json ファイルを生成できる。



図 13 OpenPose 出力動画・対象 2 人

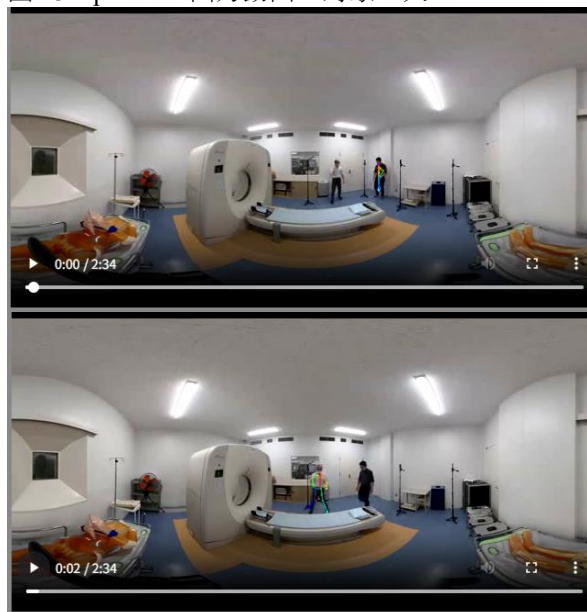


図 14. OpenPose 出力動画・異なる対象

次のような問題がある。図 14 は、OpenPose によって出力された同じ動画の 2 つの異なるフレームを示している。この場合、MocapNET は OpenPose によって出力された json ファイルの 1 人分の骨格 (2D ポーズ) データしか受け入れないため、パラメータ「?number_people_max 1」を使用している。2 つのフレームに見られるように、OpenPose は対象の人物を誤って認識することがある。そのため、重ね合わせた骨格 (2D ポーズ) の対象が異なってしまっている。これは、OpenPose を使用する際の問題の 1 つである。複数人物の骨格 (2D ポーズ) データのそれぞれの中心位置を正しく追跡することで、この問題を解決できると考えられる。他の問題として、図 12 のモーションデータは歪んでいるように見える。元々、360 度 VR 動画は歪んでいるために、この問題が発

生している。歪を解消する処理を適用する必要がある。

6 おわりに

LiDAR スキャナや360度VRカメラなど、多くの種類のスキャニングデバイスが研究開発されている。これらのデバイスを使用することで、3Dモデル、PCD：点群データ、360度VR画像/動画などの3Dデータを作成できる。このようなデータを使用して、VRツアーなどのXR教材を制作するためのフレームワークを既に提案している。その1つは、多地点360度VR動画のVRツアー用フレームワークである。そのような多地点360度VR動画において、動画内の人の動作を分析し、次に何をすれば良いか等のコメントを学習者に提供できれば、教育効率が向上する。そのための初期段階として、OpenPoseとMocapNETを使用したモーション生成のウェブサービスを開発した。本稿では、OpenPoseとMocapNETがウェブサーバ上でどのように連携して動作するのかを説明した。生成されたモーションの例を示し、現在の問題点について議論した。

今後は、既に提案しているフレームワークを使用して数多くのVRツアーを作成し、その有用性を明らかにする。また、動画に含まれる人の行動を分析し、次にどのような操作をすべきかの情報を学習者に提供する支援機能(AIアシスタント)の実装を予定している。

謝辞

本研究の一部は、九州大学データ駆動イノベーション推進本部・次世代型オープンエデュケーション推進部門の支援を受けた。

参考文献

- [1] Okada, Y., Kaneko, K., Shi, W. 2023, Development Framework Using 360-degree VR Camera Cameras and Lidar Scanners for Web-Based XR Educational Materials Supporting VR Goggles. Proc. of 11th International Conference on Emerging Internet, Data & Web Technologies (EIDWT 2023), pp. 401-412.
- [2] Okada, Y., Kaneko, K., Shi, W. 2023. Development Framework for Web-based VR Tours and Its Examples. Proc. of 27th International Conference Information Visualisation (IV 2023), pp. 420-425.
- [3] Okada, Y., Kaneko, K., Shi, W. 2023, Development Framework based on OpenVSLAM for Web-based VR Tours Using 360-degree VR Camera Videos, Proc. of International Conference on WWW/Internet 2023, pp. 109-116.

- [4] Okada, Y., Shi, W., Kaneko, K. 2024, OpenVSLAM-Based Development Framework for Web-Based VR Tours Using 360-degree VR Camera Videos and Its Extensions. Proc. of 12th International Conference on Emerging Internet, Data & Web Technologies (EIDWT 2024), pp. 31-42.
- [5] Okada, Y., Shi, W., Kaneko, K. 2024, Web Service for VR Tours of 360-degree VR Camera Images/Videos Using OpenVSLAM Server, Proc. of 2024 IEEE International Conference on Consumer Electronics-Asia(ICCE-Asia 2024), pp. 1-4.
- [6] Xu, T., Shi, W., Kaneko, K. Okada, Y. 2024, Development Framework for Web-Based Multi-Location 360VR Videos Supporting VR Goggles, Proceedings of 2024 IEEE International Symposium on Consumer Technology (ISCT-2024), pp. 112-118.
- [7] Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172-186, 1 Jan. 2021, doi: 10.1109/TPAMI.2019.2929257.
- [8] Qammar, A., & Argyros, A. A. (2019). MocapNET: Ensemble of SNN encoders for 3D human pose estimation in RGB images. British Machine Vision Conference.
- [9] Sumikura, S., Shibuya, M. and Sakurada, K., 2019, OpenVSLAM: A Versatile Visual SLAM Framework. MM '19: Proc. of the 27th ACM International Conference on Multimedia, pp. 2292-2295.
- [10] OpenPose:<https://github.com/CMU-Perceptual-Computing-Lab/openpose>
- [11] MocapNET:<https://github.com/FORTH-ModelBasedTracker/MocapNET>
- [12] video2bvh:
<https://github.com/KevinLIT/video2bvh>
- [13] video_to_bvh:
https://github.com/Dene33/video_to_bvh
- [14] video to 3DPose and bvh:
<https://github.com/HW140701/VideoTo3dPoseAndBvh>

補足

Commands 1)

```
1: cd OpenPose directory
2: bin\openposedemo.exe -video xxx.avi
--write_video result.avi --write_json output/
```

Commands 2)

```
1: ./convertOpenPoseJSONToCSV --from
/path/to/outputJSONDirectory/ --label
yourVideoFile --seriallength 12 --size 1920 1080
-o .
2: ./MocapNET2CSV --from dataset/sample.csv
--visualize --delay 30
```

ここで、1:は、複数の json ファイルから専用の csv ファイルを生成する。2: は、専用の csv ファイルから一つの bvh ファイルを生成する。

Commands 3)

```
1: md
d:\xxx\OpenPose\openpose16_gpu\output\jso
ns\${str1}
2: cd d:\xxx\OpenPose\openpose16_gpu
3: ffmpeg.exe -i input/videos/${bName} -vf
scale=640:-1 temp/videos/${str1}.avi;
4: bin\openposedemo.exe --video
temp/videos/${str1}.avi --write_video
temp/videos/result.avi --write_json
output/jsons/${str1};
5: ffmpeg.exe -i temp/videos/result.avi
output/videos/${bName};
6: lhaplus.exe /c:zip
/o:/output/jsons/ ./output/jsons/${str1};
7: wsl /home/xxx/json2bvh.sh
/mnt/d/xxx/OpenPose/openpose16_gpu/output/js
ons/${str1} ${str1}_
/mnt/d/xxx/BVHViewer/models/bvh/${str1}.bvh;
```

ここで、\${str1} は入力動画ファイルの拡張子なしファイル名、\${bName} は入力動画ファイルのファイル名である。シェルスクリプト json2bvh.sh の中身は Commands 2)と同様である。