大和大学　研究紀要　第12巻　理工学部編　2026年３月　pp.1～6

複数のアプリケーションに対してデータを送信する骨格推定システム

A Human Pose Estimation System that Sends its Data to Multiple Applications

仁王修平*　岸田卓樹*　阿部哲也*
NIO Shuhei　KISHIDA Takuju　ABE Tetsuya

要　旨

本研究では，OpenCV，YOLOv8およびMMPoseを用いて人物の骨格をリアルタイムに推定し，Socket.IOを介し複数のアプリケーションに同時配信するシステムを構築した。本システムでは，推定された骨格の関節にあたるキーポイントをアプリケーションに対して送出し，さらに，カメラ映像を送出するかどうかをアプリケーション側から選択可能とした。また，本システムを用いて，demo，Rain，Factory，Reepの４種類のインタラクティブアートを実装した。実験では，接続するクライアント数を１から20まで増加させ，それに伴うCPU，GPU利用率，送信フレームレート，ネットワークビットレートを測定した。その結果，映像送出を行わないときはクライアント数の増加に伴ってCPU，GPU使用率および送信フレームレートに有意な差は見られなかったが，映像送出を行うときはクライアント数の増加に伴ってCPU使用率は上昇し，GPU使用率および送信フレームレートは下降した。また，両条件でネットワークビットレートは接続クライアント数の増加に比例して増加した。

Abstract

In this paper, we developed a system to multicast estimated human pose data to multiple applications. Our system emplogs OpenCV, YOLOv8, and MMPose to estimate human pose and Socket.IO to multicast the data. The system streams key points corresponding to joints of the estimated human pose to the applications and allows applications to optionally receive the camera images. We developed four types of interactive art using our system: demo, Rain, Factory, Reep. In experiments, we increased the number of connected clients from 1 to 20 and measured the corresponding CPU and GPU utilization, transmission frame rate, and network bitrate. Results showed that when video was not being streamed, increasing the number of clients did not significantly affect CPU or GPU utilization or transmission frame rate. However, when video was being streamed, increasing the number of clients caused CPU utilization to rise while GPU utilization and transmission frame rate decreased. Furthermore, under both conditions, the network bitrate increased proportionally with the number of connected clients.

キーワード：骨格推定，複数アプリケーション，インタラクティブアート
Keywords：Human Pose Estimation，Multiple Applications，Interactive Media Art

１．はじめに

近年，カメラ技術や機械学習の発展により，映像から人の姿勢や動作を自動的に推定する骨格推定技術が多様な領域で応用されるようになってきた。しかし，既存のシステムでは，単一のアプリケーション内での利用を前提としており，拡張性に欠けるという欠点が存在する。

そこで本研究では，YOLOv8［1］とMMPose［2］を組み合わせた骨格推定システムを基盤とし，Socket.IO［3］を利用して複数のアプリケーションへリアルタイムに骨格データを配信できるシステムを構築した。本システムでは，データ送信先として開発者が自由に表現や機能を拡張したプログラムを指定することができる。

本研究では，複数のインタラクティブアート作品を実装し，アプリケーション数の増加がシステム性能に与える影響を評価した。

２．関連研究

本節では，本研究に関連する研究を述べる。

崔ら［4］は，複数観客の身体情報に基づくインタラクティブアート作品『未知』を製作した。『未知』は，

*大和大学理工学部
令和７年12月５日

カメラを用いて観客の位置や動作に関する情報を認識し，その情報により音響および映像を変化させるインタラクティブアート作品である。また，賈ら［5］は，モーションに応じてエフェクトが変化する体験型メディアアートの制作を行った。賈らの作品は，体験者が指定の動作を行うことでモーションと視覚効果がリアルタイムに演出される。一方我々は，カメラを用いて取得した1つの情報に基づいて，複数のシステムを動作させることができるシステムを実装した。

VP-bridge［6］は，i-PRO製のAIカメラからの映像および骨格情報をWebSocketを用いて複数のアプリケーションに対して送信する製品である。我々のシステムでは，市販のWebカメラを用いて複数のアプリケーションに対して骨格データおよび映像を送信することができる。市販のWebカメラを用いることにより，より安価にシステムを構築することが可能になる。

人間を含む物体を検出する手法には，SIFT［7］，HOG［8］，SSD［9］，DETR［10］等がある。我々は，高速かつ高精度であるとされ，モデルを変更することにより負荷を調整することができる点から，Ultralytics YOLOv8を採用した。

骨格推定を行う実装は数多く存在する［11］（例: VitPose［12］，MotionBERT［13］）が，我々は，YOLOとの接続性や複数人の骨格を推定することができる点から，MMPoseを採用した。

3．実装

本システムの構成を図1に示す。

本システムはまず，OpenCV［14］を用いてWebカメラの画像を取り込み，前処理を行う。本実装では，画像を1920x1080 30fpsで取得した。次に，YOLO を用いて物体検出を行う。YOLOは，人間（person）だけでなく様々な物体を検出することができる。そのうち人間だけを取り出し，MMPoseを用いて骨格推定を行う。推定された骨格のうち関節にあたるキーポイントをSocket.IOを用いてアプリケーションに送出する。同時に，jpeg 90％に圧縮したカメラ映像をSocket.IOを用いてアプリケーションに送信することを選択可能にした。カメラ映像を送信するかどうかを選択可能にすることで，アプリケーションの要件やデバイスに合わせてデータ量の削減や軽量化を行うことができる。

本実装には，Python 3.12.10，PyTorch 2.5.1+cu121，CUDA 12.1.66，cuDNN 9.1.0，OpenCV 4.12.0，YOLO 8.3.174，MMPose 1.3.2を用いた。

4．アプリケーション

本システムを活用するアプリケーションとして，demo, Rain, Factory, Reep を実装した。demoは本システムを検証する目的で作成されたアプリケーションであり，Rain，Factory，およびReepは本システムを用いて作成されたインタラクティブアートである。以下の節にそれぞれについて詳述する。

4.1. demo

このアプリケーションはシステムの評価や実装状況の確認を目的として，HTMLおよびJavaScriptを用いて作成された。実行画面を図2に示す。Socket.IOより受信した骨格推定データに対して，キーポイントごとにワイヤで繋ぎフレームとして表示する機能がある。さらに，キーポイントにわかりやすいラベルを付ける機能，それらの映像にカメラ映像をオーバーレイする機能があり，それぞれ画面上のトグルボタンを用いて切り替えることができる。

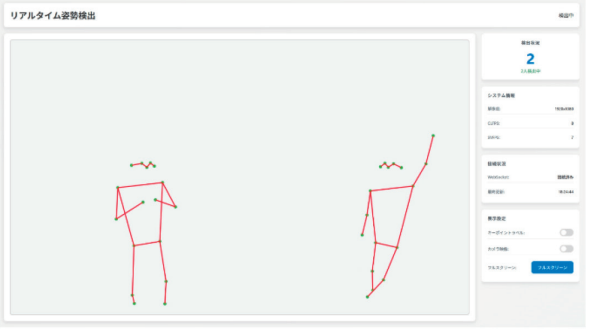

図2：demo の動作画面

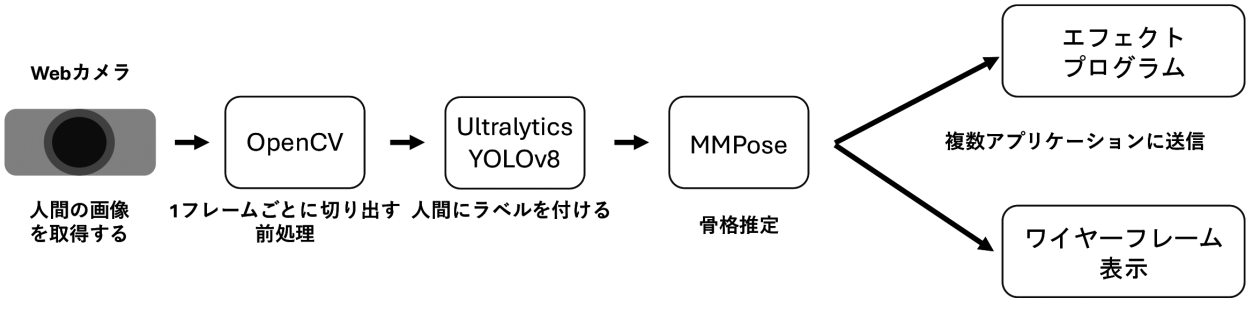

図1　本システムの構成

4.2. Rain

Rainはユーザが葉の形状をした傘により，雨から身を守る様子を表現するアプリケーションである。実行画面を図３に示す。

本アプリケーションでは骨格推定によって得られた右手首のキーポイントから右手の座標を計算し，その座標に葉の傘の画像を重ねて表示する。また，ユーザの頭部のキーポイントに花の画像を表示する。

このアプリケーションでは送られてきた骨格推定データをp5.jsとHTMLを用いて処理し，ブラウザ上に表示する。表示しているアニメーションは自作のドット絵を採用している。ユーザの頭や右手の位置が変化すると，それに追従して花や葉の傘の位置も連動して変化する。これは，骨格データをもとに仮想的な物体をインタラクティブに重ねて表示する仕組みを実現している。

図３：Rain の動作画面

4.3. Factory

Factoryは，ユーザの体を電気の導線として見立てて，仮想空間上で電球をともすような表現をするアプリケーションである。図４に実行画面を示す。

Rainと同様にユーザの頭部座標を割り出し，花の画像を表示する。また，両手首のキーポイントから両手の座標を計算し，画面上に配置されているスクリプト（歯車，パイプ，ボタン）との接触を検出すると，電球が灯るアニメーションが表示される。さらに，ユーザが両手を合わせたときに，コイルに電流が流れるアニメーションが表示される。このように，Factoryでは身体動作を用いたインタラクティブな物理現象の表現を行っている。

図４：Factory の動作画面

4.4. Reep

Reepは，画面上を移動するキャラクタであるReepを捕獲することを目標にしたアプリケーションである。図５に実行画面を示す。

Reepでは，ユーザの頭部に花が描画され，ランダムな位置にReepという名前のキャラクタが描画される。Factoryと同様に，骨格推定によって得られた両手首のキーポイントから両手の情報を推測し，ユーザが右手でReepを捕まえようとすると一度消失し，ユーザの左手の位置に再出現するアニメーションを表示する。これは，ユーザの動作に対して即時的な反応を得ることができ，キャラクタとのインタラクションをより強調するものとなっている。

図５：Reep の動作画面

５．評価

本システムは複数のアプリケーションに対して一斉に姿勢データを送信することができる。 送信先のアプリケーションの数が増加したときに，本システムの性能に与える影響を調査した。

5.1. 方法

本実験では，サーバPC，クライアントPC，動画再生用PCを用いた。実験条件の統制のため，まず，Logicool Meetup（最大解像度: 3840×2160，最大フレームレート30fps）を用いて，常に著者２名の全身が映るように映像を撮影した。撮影された映像を動画再生用PCにて再生した。再生された映像はキャプチャボードを通してサーバPCに入力され，Webカメラ映像の代わりとした。サーバPCにて本システムを稼働させ，クライアントPCにて4.1節に示したdemoアプリケーションを稼働させた。クライアントPCは５台用意され，それぞれにおいて０個から４個のアプリケーションを稼働させた。同時に稼働しているアプリケーションの総数を１個から20個まで増やしていき，サーバPCのCPU・GPU使用率，送

信データのビットレート，および送信されるデータのフレームレートを調査した。

5.2. 実験環境

本実験では，サーバPC（OS: Windows 11 Pro，CPU: Intel Core i5 11400，メモリ: 16GB，GPU: NVIDIA RTX 3080），５台のクライアントPC（OS: Windows 11 Pro，CPU: Intel Core i5 1235U，メモリ: 32GB，GPU: Intel Iris Xe Graphics），および動画再生用PCを用いた。また，通信を安定させるためデータを送信するネットワークは1Gbpsの有線接続とし，インターネットから切り離した環境で実施した。

サーバPCの実行環境には，3節にて示した環境を用いた。クライアントPCにおける実行には，Google Chrome 141を用いた。

5.3. 結果と考察

クライアントPC上で稼働させるクライアントの総数を１個から20個まで順に増やしたときのサーバPCのCPU使用率を図６に，GPU使用率を図７に，送信データのフレームレートを図８に，送信データのビットレートを図９および10に示す。

図６および７より，接続されたクライアント数が１から20の間では，カメラ映像を送出しない場合にはサーバPCのCPUおよびGPU使用率に有意な差は見られなかった。このため，カメラ映像を送出しない環境ではクライアント数が20以下の場合，クライアント数１の時と同じCPUおよびGPUを用いることができると考えられ

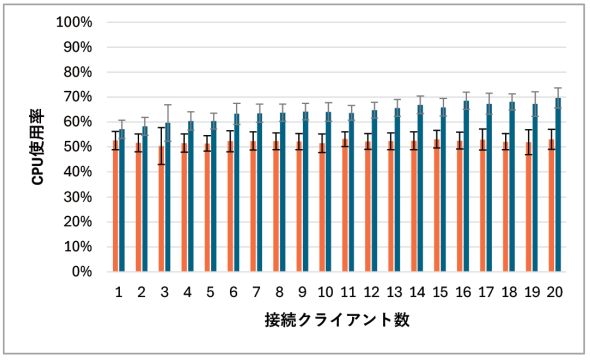

図６：接続されたクライアント数に対するサーバPCのCPU使用率［%］
橙：カメラ映像なし，青：カメラ映像あり
（エラーバーは標準偏差を表す）

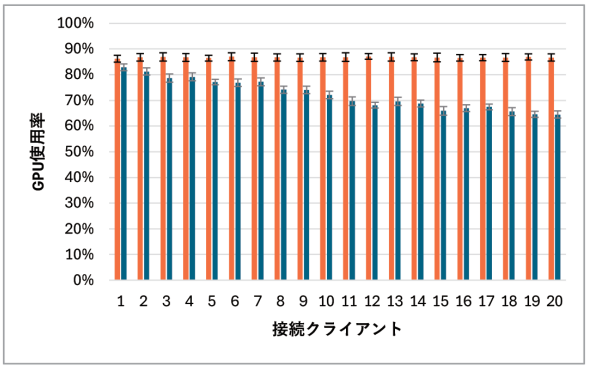

図７：接続されたクライアント数に対するサーバPCのGPU使用率［%］
橙：カメラ映像なし，青：カメラ映像あり
（エラーバーは標準偏差を表す）

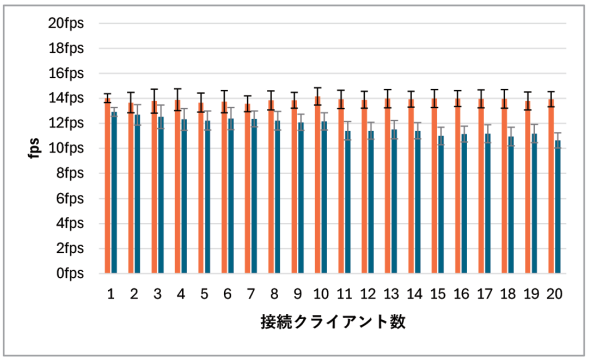

図８：接続されたクライアント数に対するサーバPCの送信データのフレームレート［fps］
橙：カメラ映像なし，青：カメラ映像あり
（エラーバーは標準偏差を表す）

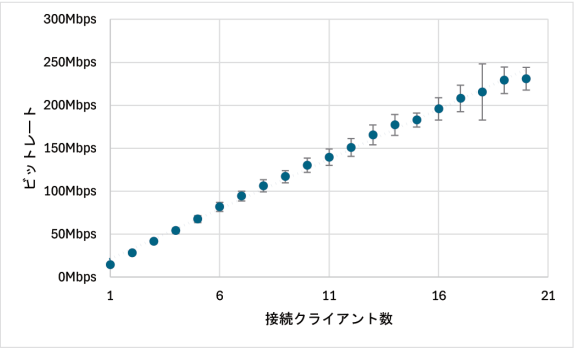

図９：カメラ映像を送出しない時のネットワークビットレート［Mbps］
（エラーバーは標準偏差を表し，点線は回帰直線を表す）

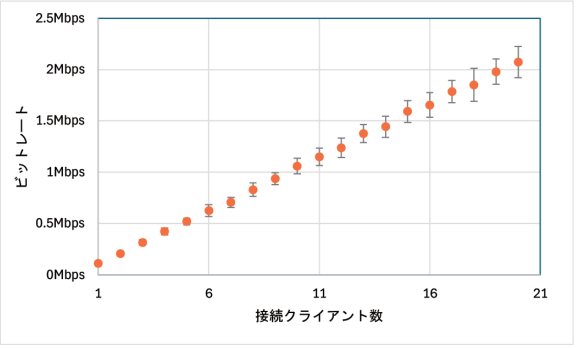

図10：カメラ映像を送出する時のネットワークビットレート［Mbps］
（エラーバーは標準偏差を表し，点線は回帰直線を表す）

る。一方，カメラ映像を送出する場合ではクライアント数を1台の場合と20台の場合を比較した際に，CPU使用率は有意に上昇し（p = 9.31E-22 < 0.01），GPU使用率は有意に下降した（p = 4.88E-38 < 0.01）。

図８より，サーバPCがカメラ映像を送出しない場合ではクライアント数増加に伴うフレームレートの変動に有意な差は見られなかった。一方，カメラ映像を送出する場合はクライアント数1の場合と20の場合を比較した際に，フレームレートは有意に下降した（p = 3.83E-36 < 0.01）。このことより，サーバPCがカメラ映像を送出する場合は各アプリケーションで必要なフレームレートを考慮してクライアント数を設定する必要がある。

図６，７および８より，接続クライアント数の増加に伴いCPU使用率が上昇し，GPU使用率およびフレームレートが下降していることがわかる。このことから，CPU，GPU以外の部分にボトルネックが存在し，そのためフレームレートが下降していることが示唆される。

図９および図10より，サーバPCがカメラ映像を送出しない時のネットワーク利用は，１クライアントあたり0.1Mbps程度，映像を送出する時のネットワーク利用は15Mbps程度であることがわかる。近年のネットワークインタフェースは１Gbpsのものが主流であるため，20クライアントまでの場合ネットワークはボトルネックにならないと考えられる。

６．今後の展望

本システムに組み込むことのできる要素としてARや機械学習による姿勢検知が挙げられる。このシステムをARと活用することによって視覚情報を超えた，没入感の高い体験を行えると考える。また，取得した骨格データを活用した高度なジェスチャ認識技術の構築と，微細な運動特性の解析に基づく情動推定モデルとの統合を進める。これにより，喜びや怒りといった感情状態を反映したマルチモーダル・インタラクションの実現を目指す。また，複数ユーザ間の相互作用を考慮したインタラクション設計を導入することで，協調的な表現形式の創出が可能になると考えられる。

さらに，リアルタイム骨格データに基づきロボットアームを制御するシステムを構築することで，デジタル信号と物理的動作を結合したハイブリッド型アート表現の可能性を検証する。 加えて，複数のデバイスへの骨格データ配信を通じて，１人の観客が空間全体に影響を与えるインタラクティブアート環境を設計することにより，従来の個別的・受動的な鑑賞形態を超え，空間全体を媒介とした新たな表現様式の確立を目指す。

また，人物の姿勢や動作の異常検知は，セキュリティシステムへの応用が大いに期待される。この複数デバイスに配信できることは従来の骨格推定システムにはない強みである。これは異常状態をAIで検知すると同時に別デバイスに配信することにより，人間とAIの二重のチェックが可能になる。その結果，従来よりも信頼性の高い運用ができると考えられる。

映像を送出する通信方式の一つにWebRTC［15］がある。本システムでは実装上の容易さからWebSocketのライブラリであるSocket.IOを採用したが，WebRTCはWebSocketに比べ映像を高速かつ高効率に転送することができる上，DataChannelを使用することにより映像と同期したタイミングで骨格推定データをクライアントに配信することができるという利点がある。通信をWebSocketからWebRTCに変更し，本システムの性能が向上するかどうか検証したい。

７．まとめ

本研究では，骨格推定技術を用いて複数のアプリケーションに対して骨格情報を配信できるシステムを構築した。YOLOv8，MMPose，Socket.IOを組み合わせることにより，カメラ映像から取得したデータからリアルタイムで姿勢検出を行い，そのデータを用いてdemo，Rain，Factory，Reepといった複数のアプリケーションを実装した。

本システムの性能を調査する実験を行い，その結果，サーバPCから映像を送出した場合，クライアント数の増加に伴ってもCPUおよびGPUの使用率には有意な変化は見られなかったが，映像を送出した場合はCPU使用率が有意に上昇し，GPU使用率は有意に下降した。また，送出するデータのフレームレートは，映像送出を行う場合，クライアント数の増加に伴い有意に下降した。

本研究はインタラクティブなアートや，セキュリティ，ロボットとの融合などの多数の分野に応用可能である。今後本研究では，ジェスチャ認識や感情推定，ARとの統合に発展させていきたい。

参考文献

［1］Muhammad Yaseen. What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. arXiv 2408.15857. 10 pages. 2024.
［2］MMPose Contributors. OpenMMLab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose（2025-11-09 閲覧）
［3］Socket.IO https://socket.io/（2025-11-09 閲覧）
［4］崔 恩宇, 三戸 勇気. 複数観客の身体情報に基づくインタラクティブアート作品「未知」. 先端芸術音楽創作学会 会報 Vol.17 No.3 pp.1–4. 2025
［5］賈 佳恵, 向野 誠, 木村 鷹丸, 川北 輝. モーションに応じてエフェクトが変化する体験型メディアアートの制

作. 第30回日本バーチャルリアリティ学会大会論文集. 3 pages. 2025年9月.
［6］ネクストシステム. VP-Bridge. https://www.next-system.com/edge-ai/vp-bridge（2025-11-09 閲覧）
［7］David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. The International Journal of Computer Vision. 28 pages. 2004.
［8］N.Dalal and B.Triggs. Histograms of Oriented Gradients for Human Detection. IEEE Computer Vision and Pattern Recognition, 886–893, 2005.
［9］Shuting Zhao, Linxin Bai, Liangjing Shao, Ye Zhang, Xinrong Chen. SSD-Poser: Avatar Pose Estimation with State Space Duality from Sparse Observations. Proceedings of the 2025 International Conference on Multimedia Retrieval. 2025.
［10］Sebastian Janampa, Marios Pattichis. DETRPose: Real-time end-to-end transformer model for multi-person pose estimation. arXiv preprint arXiv:2506.13027. 2025.
［11］Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. 2023. Deep Learning-based Human Pose Estimation: A Survey. ACM Comput. Surv.56,1,Article 11（January 2024), 37 pages. https://doi.org/10.1145/3603618
［12］Yufei Xu, Jing Zhang, Qiming ZHANG, Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. Advances in neural information processing systems 35 (2022): 38571-38584.
［13］Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. Proceedings of the IEEE/CVF international conference on computer vision. 2023.
［14］OpenCV Team. OpenCV. https://opencv.org/（2025-11-09 閲覧）
［15］WebRTC. https://webrtc.org/（2025-11-09 閲覧）