2024 10th IEEE RAS/EMBS International Conference for
Biomedical Robotics and Biomechatronics (BioRob)
1-4 September 2024. Heidelberg, Germany

2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob) | 979-8-3503-8652-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/BIOROB60516.2024.10719940

Biomechanical Arm and Hand Tracking with Multiview Markerless Motion Capture

Pouyan Firouzabadi[1,2], Wendy Murray[1,2], Anton R Sobinov, J.D. Peiffer[1,2], Kunal Shah[2], Lee E Miller[1], and R. James Cotton[2,3]

***Abstract*— Human arm and hand function is extremely complex with many degrees of freedom. It is also a common target for clinical interventions. However, precisely measuring upper extremity movement in both clinical and research settings is logistically challenging. We overcame this challenge through a novel approach to reconstructing arm biomechanics from markerless motion capture from multiple synchronized videos. Our approach directly optimizes the kinematics of an accurate biomechanical arm and hand that allows end-to-end minimization of the errors between the reconstructed movements and keypoints detected by computer vision. Key to this is an implicit function that maps from time to joint kinematics, which provides a learnable trajectory representation that can be differentiated through the biomechanical model, and supports GPU acceleration using MuJoCo-MJX. This optimization solves for the inverse kinematic solution consistent with the measured keypoints, consistent with biomechanical constraints, in addition to scaling the model while solving for the kinematics. We compare different hand keypoint detectors and find the best produces a fit with only several millimeters of reconstruction error. We also find that end-to-end optimization outperforms a two-stage fitting procedure, equivalent to more traditional biomechanical pipelines, where we first compute 3D marker trajectories and then perform inverse kinematics fitting in OpenSim. We anticipate this framework will reduce the barriers to biomechanical analysis of the arm and hand in both clinical and research settings.**

I. INTRODUCTION

Biomechanical characterization of upper extremity movements provides useful insight into neuromuscular control mechanisms and neurological dysfunction. Due to the anatomical complexity of the upper limb and the diverse functions our hands and arms perform, measuring these movements well remains challenging. Motion capture of hand movement is a fundamental component of the effective assessment of hand motor function. The standard for 3D kinematic analysis recommends high-fidelity equipment applied by experts in the technology [1]. The hand has 21 degrees of freedom (DoF), enabling intricate hand movements via the agility of the fingers and thumb, all of which introduce many challenges. For example, occlusion of individual finger motions caused by other fingers during hand motion capture is a common concern [2].

Advancements in computer vision have produced accurate human pose estimation (HPE) algorithms [3]. These methods include algorithms trained to approximate the location of 21 markers, referred to as "keypoints", corresponding to joints and fingertips in the hand in an image [4], [5], [6], [7], [8]. 2D keypoints are generally detected on a frame-by-frame basis, with no biomechanical constraints linking an identified keypoint's location between frames. Using multiple synchronized and calibrated cameras, the 2D keypoints can be triangulated into 3D space [9]. However, given the lack of biomechanical constraints between frames, triangulating at each time step independently yields suboptimal reconstructed 3D trajectories that can deviate from achievable hand motions. Prior work in gait markerless motion capture (MMC) analysis has incorporated kinematic models to explicitly constrain the triangulated 3D coordinates computed from HPE approximations [10], [11]. Having abundant HPE keypoints for gait analysis has enabled using an implicit representation that learns a mapping to 3D marker locations with additional constraints; improving the MMC trajectory representations [12], [13]. Replicating the complex biomechanical constraints of real hand movements presents novel challenges, beyond those for whole body movements during walking, the focus of these prior studies. An accurate end-to-end MMC workflow capable of tracking anatomically plausible 3D hand trajectories would be a critical advance for the field.

Here, we evaluate the efficacy of incorporating biomechanical constraints for hand motion analysis using state-of-the-art kinematic models of the hand, wrist, and arm. To accomplish this, we integrated a 21 DoF model of the hand [14] with a 7 DoF biomechanical model of the upper extremity [15]. The combined model includes: kinematic definitions of all finger and thumb movements (including coupled flexion for the carpometacarpal joints of the ring and little finger); a representation of the wrist that accounts for the non-

This work was supported by Northwestern University (NIH R01 NS131953), the Restore Center P2C (NIH P2CHD101913), and the Research Accelerator Program of the Shirley Ryan AbilityLab

[1]Department of Biomedical Engineering, Northwestern University, Evanston, IL pouyan@u.northwestern.edu

[2] Shirley Ryan AbilityLab, 335 E Erie St, Chicago, IL

[3] Department of Physical Medicine and Rehabilitation, Northwestern University, Evanston, IL rcotton@sralab.org

orthogonality of joint axes of rotation and begins to address the complex dependence of global wrist motion on carpal bone kinematics; and appropriate definitions of forearm rotation (pronation/supination), elbow and shoulder DoFs. These models were originally developed in OpenSim, a free, open-source biomechanical simulation software for kinematic and dynamic simulations [16]. To enable computing the derivatives of keypoint locations with respect to kinematic parameters and limits, we converted [17] the 28-DoF kinematic model to MuJoCo [18]. MuJoCo is a biomechanical modeling software widely adopted in machine learning research enabling the implementation of the end-to-end approach. Ultimately, implementing these models in MuJoCo-MJX [19] allows for GPU acceleration and faster analysis time by integrating features from the Brax physics simulator.

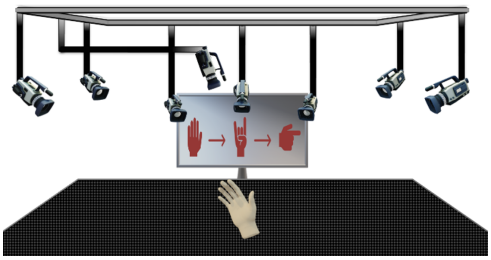
The end-to-end approach allows for a seamless processing of data from 2D keypoint detection to biomechanically constrained 3D motion reconstruction. Traditionally, reconstructing constrained 3D motion involves two stages 1) triangulation and post-processing of estimated 2D keypoints to 3D trajectories 2) scaling and solving the inverse kinematic problem on an external biomechanical model. To compare these two approaches, multiple HPE algorithms used for the hand were applied. We show that our optimized end-to-end pipeline produces significantly improved hand kinematics versus the two-stage approach regardless of the HPE method used. Specifically, our contributions are:

- Demonstrate the two-stage pipeline that fits the biomechanical model of the hand to triangulated 3D MMC data
- Perform model scaling, marker, and kinematic optimization integrated with GPU acceleration using Mujoco-MJX
- Compare two approaches using different hand pose estimation algorithms
- Show the ability of the end-to-end approach in providing biomechanical feedback while improving 3D trajectories compared to two-stage approach

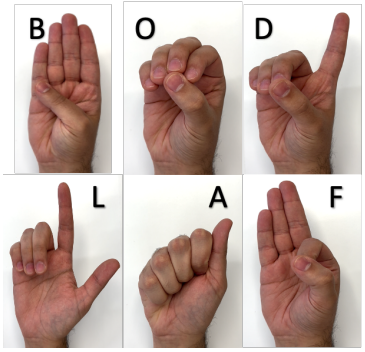
II. Methods

A. Data Acquisition

This study received approval from the Northwestern University Institutional Review Board with all participants providing an informed consent. Four individuals were seated on a chair with armrests to engage in specific hand tasks. In each task, participants maintained their arm in a rested state, with the elbow bent at a 90-degree angle. They were instructed to form the American Sign Language letters "A", "B", "D", "F", "L", and "O" as shown in Figure 1. Throughout these tasks, participants transitioned their hands from a supinated position to mid-prone and then fully pronated. We quantitatively



(a) 8 Camera Configuration



(b) 6 Tasks Performed

Fig. 1: Data acquisition setup and experimental tasks

assessed our approaches by conducting eight trials of the "A" task with four participants and qualitatively visualized the "D" and "A" signs performed by a single participant.

While performing these sequences, videos were recorded from a markerless motion capture system consisting of 8 FLIR BlackFly S GigE RGB cameras. This system records 2048 by 1536-pixel images at 30 frames per second synchronized using the IEEE1558 protocol [12], [13]. We used a mixture of F1.4/6mm and F1.6/4.4-11mm variable focus lenses, with camera positions and lenses set to keep the hand and upper body in view under the full range of motion.

The cameras were calibrated from videos of a moving 7x5 ChArucoBoard with 110mm spacing [20]. Extrinsic and intrinsic matrices were estimated using the anipose library [21]. The calibrated camera parameters produce a function denoted as Π_i for each camera i. This function projects points in 3D space onto the 2D image plane of each camera: $\Pi_i : x \rightarrow y, x \in \mathbb{R}^3, y \in \mathbb{R}^2$

B. Video Processing

We processed videos using PosePipe [22], an open-source tool for human pose estimation. We used MeTRAbs-ACAE [23], a keypoint detector trained on dozens of datasets, to extract biomechanically used arm keypoints from the MOVI keypoint set [24]. MMPose [25] was used to extract hand keypoints from the COCO-Halpe dataset [26], [27], containing 135 keypoints representing the face, body, and hands. We used these

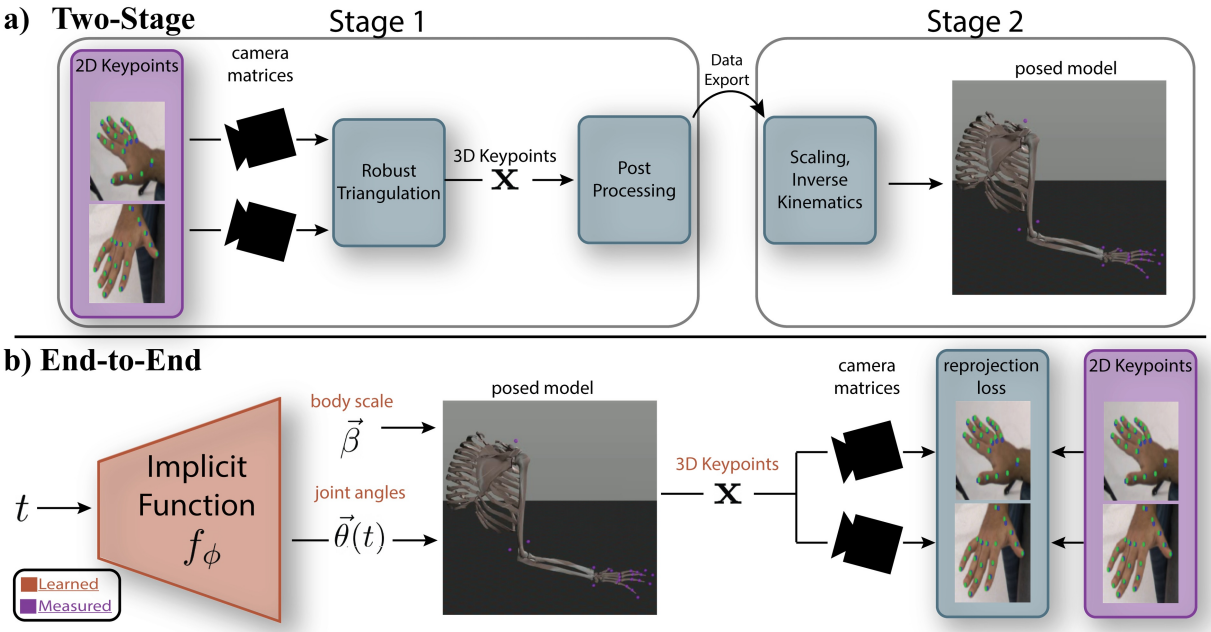


Fig. 2: Fitting methods. (a) Two-stage methods first triangulate detected 2D keypoints without biomechanical constraints and perform inverse kinematics on those fits. (b) We jointly optimize the body scale parameters and an implicit function that outputs joint angles at a given recording timestep. This process is supervised by comparing the 3D keypoint locations from the forward kinematic process with detected 2D keypoints.

keypoints to localize each hand before applying the hand-specific keypoint detectors described next.

C. Hand Keypoint Detection

A bounding box was computed for the right hand using the hand keypoints from Halpe, followed by an additional 60-pixel dilation. Using the MMPose library [25], we applied multiple hand-specific keypoint detection algorithms to this cropped image, detailed in Table I, allowing for comparison across out-of-the-box publicly accessible HPE architectures.

TABLE I: 2D HPE algorithms and EPE for datasets

Method	Dataset	EPE_{pixels}
RTMPose[4]	Hand5*	5.06
RTMPose	COCO_wholeBody_hand	4.51
Freihand [5]	Freihand2d (Multi-view)	3.27
HRNetV2 with UDP[8]	OneHand10k	23.88

*Hand5 [28] is a combination of five datasets, COCO-Whole body [29], FreiHand2d [30], OneHand10k [31], RHD2d [32], and Halpe datasets.

These models all output 21 total keypoints on each hand: the fingertips (TIP) of each finger, proximal and distal interphalangeal (PIP, DIP) joints, metacarpophalangeal (MCP) joint of each finger, carpometacarpal (CM) joint of the thumb, and the wrist. The average end-point error (EPE) of trained models was included as a reference to how well each HPE algorithm performs on their given dataset.

We combined the 21 hand keypoints from each algorithm with a subset of the MOVI keypoints from MeTRAbs-ACAE, to allow tracking of the arm and torso. These additional keypoints correspond to the medial and lateral wrist, medial and lateral elbow, shoulder, clavicle, and sternum.

D. Reconstruction Overview

We compare two methods to recover hand biomechanics from detected keypoints (Figure 2). The first method uses a two-stage approach analogous to traditional biomechanics that first reconstructs marker trajectories and then performs model-constrained inverse kinematic fitting to these trajectories. This is similar to earlier versions of our whole-body MMC pipeline [12], [13], although this case uses OpenSim for the inverse kinematics [16].

The second method follows our recent work using differentiable biomechanics for whole-body tracking [33]. In brief, this method jointly optimizes the body scale parameters and an implicit function f_ϕ that learns the movement trajectory constrained by a differentiable biomechanical model [34]. We have found this method to outperform the traditional two-stage approach when applied to gait analysis.

E. Two-stage Reconstruction

1) Keypoint Triangulation: This approach first triangulates 2D keypoints detected from each camera to find virtual 3D marker trajectories using the known camera intrinsic and extrinsic properties. The most common approach to calculating 3D world coordinates from 2D

locations in the camera frame is the Direct Linear Transformation (DLT) [35]. However, this approach is sensitive to outliers and occlusion caused by other fingers or limbs in a certain view. Triangulation can be made more robust by discounting keypoints from cameras that are inconsistent with the others [36]. This robust triangulation approach first triangulates many 3D coordinates using all pairs of cameras $\binom{N_c}{2}$, where N_c is the number of cameras [36]. The distance to the geometric median location is computed for each of these estimates, to identify any camera-associated outliers, in which case they are down-weighted in the DLT equations. Keypoints with low confidence are also down-weighted. Then, a weighted DLT is solved with SVD to compute the final 3D coordinates for each marker. We previously showed that this robust approach is more accurate for MMC than a standard DLT [12].

2) Model Scaling and Fitting: Next, these 3D markers are exported to OpenSim which performs model scaling and inverse kinematics, obtaining model-constrained 3D poses and keypoint trajectories. We prepared the MoBL-ARMS integrated arm [15] and hand [14] model for inverse kinematic fitting to this data by adding 28 markers, corresponding to the 21 hand keypoints and 7 MOVI keypoints noted above. Triangulated keypoints were manipulated to generate OpenSim-compatible trace files and setup files [16]. Processing keypoints for inverse kinematics (IK) optimization in OpenSim required trajectory smoothing with a median filter of size 5, interpolating over low confident keypoints, and manually creating marker to keypoint mappings. The scaling tool, referencing a static pose from the trace file, scales body segments and shifts marker locations on the model. IK optimization minimizes the root mean squared error between 3D keypoints and scaled model markers while applying biomechanical constraints. This data processing was automated, however, exported data was manually processed through the OpenSim tools performing scaling and inverse kinematic optimization to achieve the best results. The final constrained marker locations outputted by OpenSim were considered for comparison to our end-to-end approach.

F. End-to-end Reconstruction

1) Implicit Function: Based on recent work employing an implicit function triangulating 2D markers and biomechanical poses, we use an implicit function

$$f_\phi : t \rightarrow \vec{\theta} \tag{1}$$

representing a mapping from recording time t to hand poses $\vec{\theta}(t)$, implemented as a multi-layer perceptron. This function is trained to produce joint angles that are consistent with the movement of a single recording.

2) Differentiable Body Model: To allow differentiation and end-to-end optimization of this entire process, we converted our OpenSim model to MuJoCo [18] using MyoConverter [17]. MuJoCo has recently released MuJoCo-MJX which enables massive parallelization and acceleration of this process and integration with machine learning frameworks like Jax [37].

The forward kinematic process of our biomechanical model can be denoted as:

$$\vec{x} = \mathcal{M}(\vec{\theta}, \vec{\beta}) \tag{2}$$

where $\vec{\theta} \in \mathbb{R}^{47}$ are the joint angles of our biomechanical model and $\vec{\beta} \in \mathbb{R}^{4+28*3}$ is a vector that controls model scaling and keypoint offsets. This outputs the 3D constrained marker locations $\mathbf{x} \in \mathbb{R}^{28\times3}$. To evaluate the 3D keypoints at a certain time in the recording t, we simply pass the output of our implicit function (Eq. 1) at time t through this forward kinematic process as:

$$\vec{x} = \mathcal{M}(f_\phi(t), \vec{\beta}) \tag{3}$$

3) Model Scaling: MuJoCo does not provide native support for model scaling or marker offsets, which are present in traditional biomechanics software, like OpenSim [16]. While this could be done by programmatically modifying the MuJoCo XML file, this change would not be differentiable and thus not efficiently optimized. We used our prior approach [33], which uses a wrapper around the forward kinematics process that accepts isotropic scale parameters and marker offsets. In this work, we included scale parameters for the humerus, ulna, radius, and opisthenar area (top of the hand) including the metacarpal bones. We also include an overall scaling parameter, that could capture finger length differences. Combined, we represent these body and marker scaling parameters as a vector $\vec{\beta} \in \mathbb{R}^{4+28*3}$.

4) Loss Functions: To train the implicit function to follow a specific recording trajectory, we evaluate equation 3 at the recording timestamps and compare its estimate to detected 2D keypoints. Specifically, the 3D keypoint locations are passed through the calibrated camera models, Π_i, which reproject these 3D marker locations onto the 2D image plane. The location of each keypoint j, at timestep t reprojected through camera c are denoted as, $p_{t,j,c}$. The MLP is then optimized using a loss function measuring the difference between the reprojected 3D locations and 2D keypoints from HPE:

$$L_\Pi = \frac{1}{T \cdot J \cdot C} \sum_{T,J,c \in C} w_{c,t,j}\, g\left(||\Pi_c \mathbf{x}_{t,j} - p^{2d}_{t,j,c}||\right)$$

$g(\cdot)$ is the Huber loss and $w_{c,t,j}$ is the confidence of the detected keypoint. We also define an L2 regularization loss on the components of β corresponding to the model markers offsets, which prevents them from shifting much

from the initial anatomical locations:

$$L_\beta = \frac{1}{28 \times 3} \sum_{i=28*3} \beta_i^2$$

The total loss is then: $L = L_\Pi + \lambda_\beta L_\beta$

5) Optimization and Implementation Details: This model was run for 40000 iterations to intentionally overfit the pose parameters and find the best scaling parameters with the lowest loss value. Our implicit representation was implemented in Jax using the Equinox framework [38]. The MLP had layer sizes of 128, 256, 512, 1024, 2048, 2048, 4096. Time passed into the MLP goes through sinusoidal positional encoding with an encoding dimension of 17 [39], [40], and time is prescaled from 0 to π to prevent aliasing. We performed optimization with the AdamW optimizer from Optax [41], [42], using $\beta_1 = 0.8$ and weight decay of $1e-5$. The learning rate included an exponential decay from an initial value of $1e{-}4$ to an end value of $1e{-}7$. The λ for site offsets was set to $1e3$. We typically performed this on an A6000 and found it could perform more than 150 iterations a second for a single trajectory of 30 seconds.

G. Performance Metrics

Our performance metrics for comparing the two reconstruction approaches and keypoint detectors were geometric consistency (GC) and Mean Per Joint Position Error (MPJPE).

1) Geometric Consistency: To calculate the geometric consistency between all approaches, the error between the 2D keypoints and reprojected keypoints is measured in pixels. The fraction of points below a threshold number of pixels, conditioned on being greater than the specified confidence interval λ quantifies the geometric consistency (GC).

$$\delta_{t,j,c} = ||\Pi_c \mathbf{x}_{t,j} - y_{t,j,c}||$$

$$q(d, \lambda) = \frac{\sum(\delta_{t,j,c} < d)(w_{t,j,c} > \lambda)}{\sum w_{t,j,c} > \lambda}$$

We report $GC_d = q(d, 0.5)$ where $d = 10$ pixels, $\delta_{t,j,c}$ is the pixel error of the reprojected and detected keypoints, and $w_{t,j,c}$ is the confidence of the projected point.

2) Error in World Coordinates: Pixel error can be transformed into the equivalent 3D spatial error based on the camera model and distance from the camera. Specifically, the pixels of error are proportional to the spatial error tangential to the camera axis, where the proportionality coefficient scales linearly with distance from the camera. Having calibrated cameras with a known field of view f_c, the Mean Per Joint Position Error (MPJPE) for each camera can be calculated by finding the angular changes per pixel and spatial error [9]. The angular changes per pixel for each x and y axes

Fig. 3: Integrated model overlayed on all views during "D" American sign language task. Blue dots represent marker locations.

are calculated as

$$\Delta\theta^{x,y}_{t,j,c} = \delta^{x,y}_{t,j,c} \times (f_c / resolution_{x,y})$$

where $\delta^{x,y}_{t,j,c}$ represents each keypoint j's pixel error at time t for camera c across the x- or y-axis. Using the distance of the camera to each 3D keypoint in the world, D_j, spatial error is calculated. MPJPE is represented as the mean magnitude of spatial error in 3D represented:

$$\text{MPJPE}_c = \frac{1}{J \times T} \sum(||D_j \times tan(\Delta\theta^{x,y}_{t,j,c})||)$$

The median of the MPJPE across all cameras in millimeters is used for analysis.

III. Results

A. Qualitative

We visually verified the quality of the biomechanical reconstructions and 3D keypoint estimates by creating videos that reproject the 3D model and keypoints onto each camera. These showed close agreement for our end-to-end approach. Fig.3 shows an example of a participant signing the letter "D", using our best-performing approach which we describe below. In our data collection, participants signed a letter while supinated, moved to a mid-prone position, and then fully pronated their hand. An example of the movement and joint estimates during this motion can be seen from the pronator/supinator plus the metacarpophalangeal (MCP) joint angles outputted by the MJX model for letter "A" in Fig.4.

B. Performance metrics

We quantified the MPJPE and geometric consistency of both approaches as an average over our four participants signing the letter "A" three times in two separate trials. The two-stage approach, could not converge to

TABLE II: Performance metrics measurements of a trial across various tasks

Datasets	↓ MPJPE (mm)			↑ Geometric Consistency$_{q(10,0.5)}$		
	Two-Stage	End2End	Δ (mm)	Two-Stage	End2End	Δ
RTM_Hand5	4.467	**3.262**	-1.205	0.669	**0.936**	+39.9%
RTM_COCO	6.274	5.026	-1.248	0.720	0.898	+24.7%
Freihand	4.233	3.402	-0.831	0.667	0.876	+31.3%
HRNet_10kHand	6.554	5.206	-1.348	0.537	0.858	+59.7%
Halpe$_{Alpha}$	6.812	5.885	-0.927	0.336	0.514	+52.9%
Average	5.668	4.574	-1.094	0.586	0.817	+39.4%

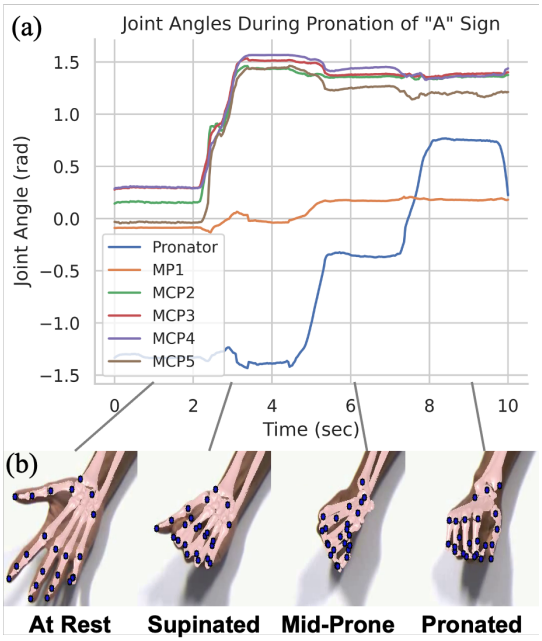


Fig. 4: MCP and Pronator joint angles in (a) corresponding to the ”A” sign performed in (b)

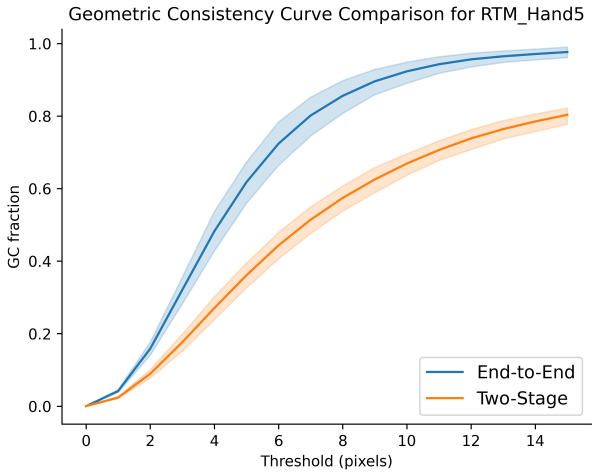


Fig. 5: Comparison of the GC curve across 4 participants with 95% CI

a solution for specific trials due to low-confident keypoints, high computational complexity, and the inability to shift markers iteratively. We compared the two biomechanical reconstruction methods using multiple HPE algorithms over the valid trials of signing letter ”A” to identify the best combination for precise hand tracking in both approaches. Since we focused on the hand, metrics were only computed for the 21 keypoints on the right hand. The Halpe dataset, providing both hand and whole body keypoints, was included as a comparison to hand-specific pose estimation algorithms.

Table II shows the median MPJPE across all cameras, and GC_{10} over 10 pixels for five HPE datasets used in both approaches. To represent the improvement by our end-to-end approach, Δ is shown for each methodology and as an overall average. Our best performance came from the end-to-end reconstruction using keypoints detected from an RTMPose network architecture [4] trained on the Hand5 dataset with a 39.9% higher geometric consistency over 10 pixels and an average of 1.205mm improvement in mean per joint position error. On average, the end-to-end approach was able to produce 39.4% higher geometrically consistent trajectories, and 1.094 millimeters lower spatial error when compared to the two-stage approach.

Plotting the geometric consistency for both approaches over all ranges of threshold for four participants is shown in Fig. 5. We observe the ability of the end-to-end approach to achieve above 0.8 consistency below 7 pixels while the two-stage achieves it over 14. The end-to-end approach, in addition to anatomically constraining the keypoints, generates marker locations with higher geometric consistency when compared to a two-stage reconstruction. Performing a paired t-test between the two approaches for both metrics shows a significant improvement in our end-to-end approach.

IV. Discussion

A fully differentiable forward kinematic model accelerated on a GPU opens up similar opportunities for hand tracking that we have seen with whole-body tracking [33]. Specifically, it enables a unified approach for end-to-end optimization of kinematic trajectories allowing

the information from the different keypoint positions and biomechanical constraints to be better combined than a two-stage approach. This is reflected in the lower sensitivity of the end-to-end reconstruction to the specific keypoint detector. It also allows jointly estimating the model scaling and marker offsets with the inverse kinematics, called bilevel optimization introduced by [10] providing a more straightforward workflow than a two-stage approach. Here, we showed the ability of implicit representation to fit biomechanically consistent trajectories to estimated keypoints while having significantly lower projection error compared to the two-stage approach.

This work has several limitations. It only evaluated able-bodied individuals performing a few movements with their hands and wrists. Additionally, this model only included the right single arm and a thorax. Finally, it only tested a specific camera configuration. There would be significant value in extending this to tracking the whole body and hands at different spatial scales. Furthermore, our work with whole-body tracking shows that a single set of scales and marker offsets can be used when fitting a set of trajectories of an individual. This can also be done over many people to optimize the base biomechanical model. These techniques would naturally generalize to this work [33]. Early studies using a whole body and hand model in development for MyoSuite of wheelchair users during wheelchair propulsion suggest this is a promising avenue. Other limitations arise from the existing HPE datasets and algorithms. The keypoints do not exactly localize to the joint centers when viewed from different perspectives. Thus there is an opportunity to develop more anatomically grounded datasets and detection algorithms.

This work moves us forward in the direction of better skeletal modeling (capturing full ranges of motion of the hand), comparison with state-of-the-art motion capture modalities, and bridging the gap between musculoskeletal biomechanical analysis of the hand in markerless motion capture systems. Recent work with object interactions and the generation of hand meshes allows for the study of hand grasps and contact estimations in a controlled environment when combined with our end-to-end pipeline [43].

V. Conclusion

Fully differentiable biomechanical models optimized on a GPU and used for end-to-end reconstruction of markerless motion capture data are promising tools for rehabilitation and movement science. In this work we showed this can be applied to enable hand tracking with an average reconstruction error in the hand of only several millimeters, using only 8 cameras. We anticipate this will improve access to detailed hand tracking, which will empower research into hand function and monitoring with rehabilitation.

References

[1] T. V. Criekinge, C. Heremans, J. Burridge, J. E. Deutsch, U. Hammerbeck, K. Hollands, S. Karthikbabu, J. Mehrholz, J. L. Moore, N. M. Salbach, J. Schröder, J. M. Veerbeek, V. Weerdesteyn, K. Borschmann, L. Churilov, G. Verheyden, and G. Kwakkel, "Standardized measurement of balance and mobility post-stroke: Consensus-based core recommendations from the third stroke recovery and rehabilitation roundtable," 2 2023.

[2] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," Computer Vision and Image Understanding, vol. 108, pp. 52–73, 10 2007.

[3] C. E. Zheng, W. Wu, C. Chen, M. Shah, C. Zheng, T. Yang, S. Zhu, J. Shen, and N. Kehtarnavaz, "Deep Learning-Based Human Pose Estimation: A Survey," Tsinghua Science and Technology, vol. 24, pp. 663–676, dec 2020.

[4] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen, "RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose," mar 2023.

[5] B. Xiao, H. Wu, and Y. Wei, "Simple Baselines for Human Pose Estimation and Tracking,"

[6] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep High-Resolution Representation Learning for Visual Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, pp. 3349–3364, aug 2019.

[7] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-Aware Coordinate Representation for Human Pose Estimation," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 7091–7100, oct 2019.

[8] J. Huang, Z. Zhu, F. Guo, and G. Huang, "The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 5699–5708, nov 2019.

[9] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge University Press.

[10] K. Werling, M. Raitor, J. Stingel, J. L. Hicks, S. Collins, S. L. Delp, and C. K. Liu, "Rapid bilevel optimization to concurrently solve musculoskeletal scaling, marker registration, and inverse kinematic problems for human motion reconstruction," p. 2022.08.22.504896.

[11] S. D. Uhlrich, A. Falisse, Łukasz Kidziński, J. Muccini, M. Ko, A. S. Chaudhari, J. L. Hicks, and S. L. Delp, "Opencap: Human movement dynamics from smartphone videos," PLOS Computational Biology, vol. 19, p. e1011462, 10 2023.

[12] R. J. Cotton, A. Cimorelli, K. Shah, S. Anarwala, S. Uhlrich, and T. Karakostas, "Improved Trajectory Reconstruction for Markerless Pose Estimation," Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference, vol. 2023, pp. 1–7, mar 2023.

[13] R. J. Cotton, A. DeLillo, A. Cimorelli, K. Shah, J. D. Peiffer, S. Anarwala, K. Abdou, and T. Karakostas, "Markerless Motion Capture and Biomechanical Analysis Pipeline," mar 2023.

[14] D. C. McFarland, B. I. Binder-Markey, J. A. Nichols, S. J. Wohlman, M. de Bruin, and W. M. Murray, "A Musculoskeletal Model of the Hand and Wrist Capable of Simulating Functional Tasks," bioRxiv, p. 2021.12.28.474357, dec 2021.

[15] K. R. Holzbaur, W. M. Murray, and S. L. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," Annals of Biomedical Engineering, vol. 33, pp. 829–840, jun 2005.

[16] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard, S. R. Hamner, E. M. Arnold, J. R. Yong, S. K. Lakshmikanth, M. A. Sherman, J. P. Ku, and S. L. Delp, "OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study

human and animal movement," PLOS Computational Biology, vol. 14, p. e1006223, jul 2018.
[17] A. Ikkala and P. Hämäläinen, "Converting Biomechanical Models from OpenSim to MuJoCo," Biosystems and Biorobotics, vol. 28, pp. 277–281, jun 2020.
[18] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," IEEE International Conference on Intelligent Robots and Systems, pp. 5026–5033, 2012.
[19] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax – A Differentiable Physics Engine for Large Scale Rigid Body Simulation," jun 2021.
[20] G. An, S. Lee, M.-W. Seo, K. Yun, W.-S. Cheong, and S.-J. Kang, "Charuco board-based omnidirectional camera calibration method," Electronics, vol. 7, p. 421, 12 2018.
[21] P. Karashchuk, K. L. Rupp, E. S. Dickinson, S. Walling-Bell, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: A toolkit for robust markerless 3D pose estimation," Cell Reports, vol. 36, p. 109730, sep 2021.
[22] R. J. Cotton, R. Org, and S. R. Abilitylab, "PosePipe: Open-Source Human Pose Estimation Pipeline for Clinical Research," arXiv, p. arXiv:2203.08792, mar 2022.
[23] I. Sarandi, A. Hermans, and B. Leibe, "Learning 3D Human Pose Estimation from Dozens of Datasets using a Geometry-Aware Autoencoder to Bridge Between Skeleton Formats," Proceedings - 2023 IEEE Winter Conference on Applications of Computer Vision, WACV 2023, pp. 2955–2965, dec 2022.
[24] S. Ghorbani, K. Mahdaviani, A. Thaler, K. Kording, D. J. Cook, G. Blohm, and N. F. Troje, "MoVi: A large multi-purpose human motion and video dataset," PLOS ONE, vol. 16, p. e0253157, jun 2021.
[25] M. Contributors, "Openmmlab pose estimation toolbox and benchmark." https://github.com/open-mmlab/mmpose, 2020.
[26] Y.-L. Li, L. Xu, X. Liu, X. Huang, Y. Xu, S. Wang, H.-S. Fang, Z. Ma, M. Chen, and C. Lu, "Pastanet: Toward human activity knowledge engine," in CVPR, 2020.
[27] H. S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y. L. Li, and C. Lu, "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, pp. 7157–7173, nov 2022.
[28] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8693 LNCS, pp. 740–755, may 2014.
[29] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, "Whole-Body Human Pose Estimation in the Wild," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12354 LNCS, pp. 196–214, 2020.
[30] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox, "FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images," Proceedings of the IEEE International Conference on Computer Vision, vol. 2019-Octob, pp. 813–822, sep 2019.
[31] Y. Wang, C. Peng, and Y. Liu, "Mask-Pose Cascaded CNN for 2D Hand Pose Estimation from Single Color Image," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, pp. 3258–3268, nov 2019.
[32] C. Zimmermann and T. Brox, "Learning to Estimate 3D Hand Pose from Single RGB Images," Proceedings of the IEEE International Conference on Computer Vision, vol. 2017-Octob, pp. 4913–4921, may 2017.
[33] R. J. Cotton, "Differentiable Biomechanics Unlocks Opportunities for Markerless Motion Capture."
[34] R. J. Cotton, A. DeLillo, A. Cimorelli, K. Shah, J. Peiffer, S. Anarwala, K. Abdou, and T. Karakostas, "Optimizing trajectories and inverse kinematics for biomechanical analysis of markerless motion capture data," in 2023 International Conference on Rehabilitation Robotics (ICORR), pp. 1–6, 2023.
[35] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision (Cited by: 11343)," Cambridge University Press, vol. 2, no. 2, p. 672, 2004.
[36] S. K. Roy, L. Citraro, S. Honari, and P. Fua, "On Triangulation as a Form of Self-Supervision for 3D Human Pose Estimation," tech. rep.
[37] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.
[38] P. Kidger and C. Garcia, "Equinox: neural networks in JAX via callable PyTrees and filtered transformations," Differentiable Programming workshop at Neural Information Processing Systems 2021, 2021.
[39] J. Zheng, S. Ramasinghe, and S. Lucey, "Rethinking positional encoding," 7 2021.
[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 2017-December, pp. 5999–6009, jun 2017.
[41] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," 7th International Conference on Learning Representations, ICLR 2019, nov 2017.
[42] J. Godwin*, T. Keck*, P. Battaglia, V. Bapst, T. Kipf, Y. Li, K. Stachenfeld, P. Veličković, and A. Sanchez-Gonzalez, "Jraph: A library for graph neural networks in jax.," 2020.
[43] C. Pokhariya, I. N. Shah, A. Xing, Z. Li, K. Chen, A. Sharma, and S. Sridhar, "Manus: Markerless grasp capture using articulated 3d gaussians," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2197–2208, June 2024.
[44] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou, and P. Daras, "A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition," IEEE Transactions on Multimedia, vol. 24, pp. 1750–1762, 2022.
[45] A. El Kaid and K. Baïna, "A Systematic Review of Recent Deep Learning Approaches for 3D Human Pose Estimation," dec 2023.
[46] D. A. Winter, "Biomechanics and Motor Control of Human Movement: Fourth Edition," Biomechanics and Motor Control of Human Movement: Fourth Edition, pp. 1–370, sep 2009.