IMUPoser: Full-Body Pose Estimation using IMUs in Phones, Watches, and Earbuds

Vimal Mollyn
vmollyn@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Riku Arakawa
rarakawa@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Mayank Goel
mayank@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Chris Harrison
chris.harrison@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Karan Ahuja
kahuja@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

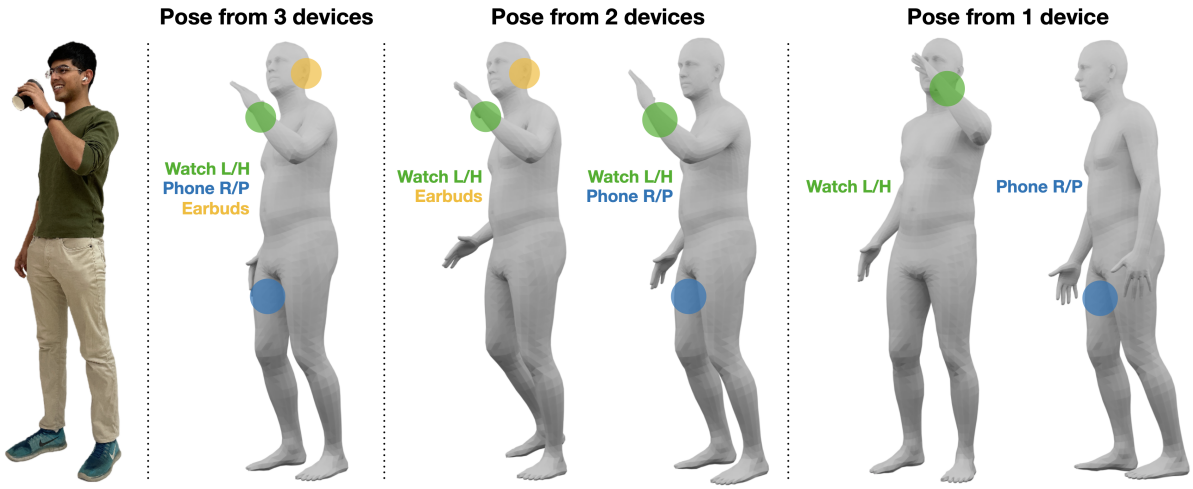

Figure 1: Using whatever mobile devices a user has with them, IMUPoser estimates full-body pose. In the best case, a user can have a smartphone, smartwatch and earbuds (pose from 3 devices). Of course, the number of devices will vary over time, e.g., earbud use is intermittent and not everyone wears a smartwatch. This means IMUPoser must track what devices are present, where they are located, and use whatever IMU data is available. Abbreviation key: L-Left, R-Right, H-Hand, and P-Pocket.

ABSTRACT

Tracking body pose on-the-go could have powerful uses in fitness, mobile gaming, context-aware virtual assistants, and rehabilitation. However, users are unlikely to buy and wear special suits or sensor arrays to achieve this end. Instead, in this work, we explore the feasibility of estimating body pose using IMUs already in devices that many users own — namely smartphones, smartwatches, and earbuds. This approach has several challenges, including noisy data from low-cost commodity IMUs, and the fact that the number of instrumentation points on a user's body is both sparse and in flux. Our pipeline receives whatever subset of IMU data is available, potentially from just a single device, and produces a best-guess pose. To evaluate our model, we created the IMUPoser Dataset, collected from 10 participants wearing or holding off-the-shelf consumer devices and across a variety of activity contexts. We provide a comprehensive evaluation of our system, benchmarking it on both our own and existing IMU datasets.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**.

KEYWORDS

Motion capture, sensors, inertial measurement units, mobile devices

ACM Reference Format:
Vimal Mollyn, Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. 2023. IMUPoser: Full-Body Pose Estimation using IMUs in Phones, Watches, and Earbuds. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3544548.3581392

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3581392

1 INTRODUCTION

Full-body motion capture is commonplace in movie visual effects and is slowly entering the consumer realm in areas such as virtual reality. Full-body pose tracking has obvious applications in gaming [38], fitness [28], rehabilitation [39], life logging [24], and context-aware interfaces [1, 5]. For example, digital assistants with knowledge of pose could help a football player improve their form, or a patient recovering from surgery monitor changes in their gait. However, at present, most consumers have no tools to track their pose, nor do they want to retrofit sensors into their homes or wear special-purpose suits or accessory devices. However, if it was possible to generate useful pose information from devices users already own, it could have a significant impact.

Most computing devices we carry with us contain IMUs, most notably smartphones, smartwatches, and wireless earbuds. In this work, we study how we can use this ecosystem of worn and mobile devices to estimate a user's body pose in real-time and with no external infrastructure. This approach introduces new challenges prior sparse IMU pose models (e.g., [21, 60]) have not faced. Uniquely, the position and number of tracked body locations can change on the go. For instance, a user can take a phone from their left pocket into their right hand, or a user can add to the number of sensed points by wearing their earbuds. Our model must accept various combinations of incomplete inputs and gracefully degrade as the number of active devices reduces (potentially to one). Secondarily, our system must work with IMU data received from consumer devices that are noisier than professional-grade motion capture suits (e.g., XSens [63]) used in highly-related prior work such as Sparse Internal Poser [60], Deep Inertial Poser [21], and TransPose [65]. Table 1 provides an overview of the most related prior work.

To evaluate our method, we created a novel dataset: professional-grade Vicon optical tracking paired with commodity device IMU data from common worn/held locations. Unsurprisingly — given that we have *at most* three body positions to estimate hundreds of degrees of freedom in the human body – our pose output is an approximation. However, it is rarely wildly incorrect, and most often, the main gestalt of a user's pose and locomotion mode is captured. This "low-fi" pose output is ill-suited for high-fidelity applications, such as special effects motion capture or virtual reality avatars, where users expect a mostly-faithful body representation. Nonetheless, the adaptive and mobile nature of IMUPoser enables passive and longitudinal sensing of the user (potentially even from a single device), making it especially well-suited for health and wellness applications. For example, this low-fi body tracking could be valuable for boosting accuracy in calorie counting, tracking progress in a physical therapy regime, and monitoring exercise form and rep count. We highlight some example uses in Figure 2.

2 RELATED WORK

We now review the related work in the area of full-body digitization. We look at both external and worn capture systems, and then review IMU-based pose-sensing approaches that are most related to our work. For an in-depth review of past and current approaches for pose estimation, we refer readers to surveys by Desmarais et al. [15] and Nguyen et al. [43].

2.1 Body Capture using External Sensors

There exists a wide range of approaches and solutions to estimate users' pose using external sensors. Commercial systems such as Vicon [58] and OptiTrack [41] use specialized hardware, including high-speed infrared cameras that track retroreflective markers attached to users' whole body or individual parts, such as the face or hands. After a calibration procedure, these systems can track large spaces at millimeter accuracy. These approaches are often used for movies, games, and character animation. The cost and setup requirements, however, preclude most consumer applications.

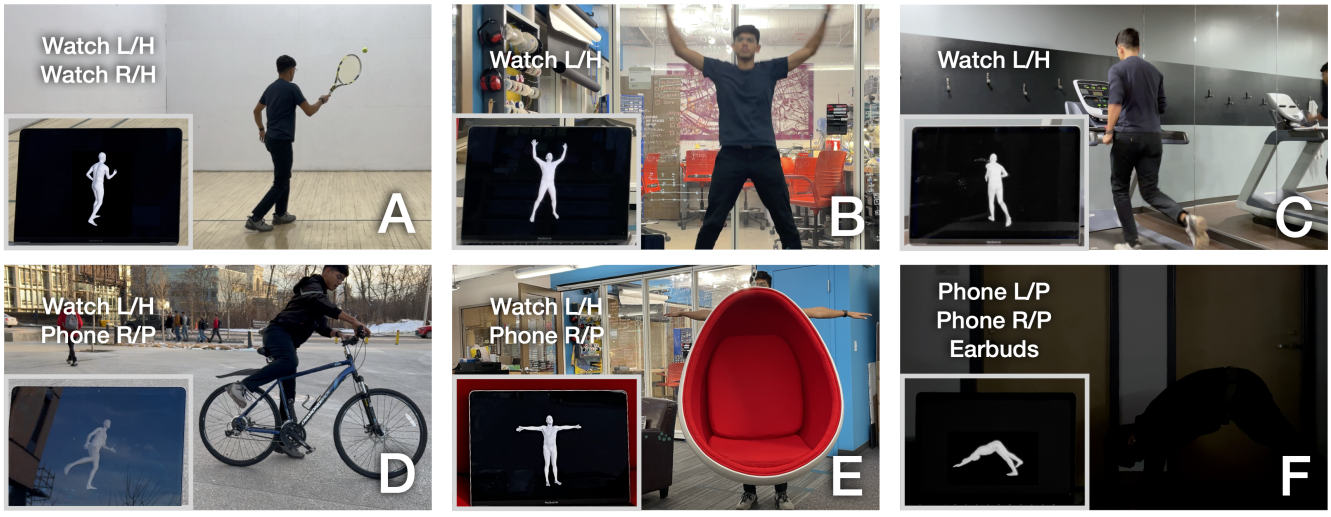

Figure 2: Real-time pose estimation (inset photos) powered by consumer mobile devices (listed in each photo) could have uses across many domains, including sports (A), rehabilitation (B), fitness (C), and transportation (D). Note also that IMUPoser is robust to occlusion (E) and lighting conditions (F). Abbreviation key: L-Left, R-Right, H-Hand, and P-Pocket.

System	# Inst. Joints	Sensor FPS (Hz)	Consumer Device	Real-time	MPJVE (cm)
XSens [63]	17	120	✗	✗	-
SIP [60]	6	60	✗	✗	7.71
DIP [21]	6	60	✗	✓	8.96
Transpose [65]	6	90	✗	✓	7.09
PIP [64]	6	60	✗	✓	5.95
Tautges et al. [56]	4	25	✗	✗	-
IMUPoser (our work)	1–3	25	✓	✓	12.08

Table 1: Comparison of worn-IMU, full-body pose estimation systems. MPJVE is calculated on the DIP-IMU dataset [21].

Current approaches for whole-body pose estimation in *consumer* applications typically rely on cheaper sensors and require less calibration. Depth cameras such as the Microsoft Kinect [45, 53, 61] and Intel RealSense [22] provide sufficient pose accuracy using medium-cost sensors for applications in VR and gaming. Zimmermann et al. [71] and Michael et al. [37], for example, extend these approaches and combine depth imagery with RGB data for improved accuracy. Such commercially-available sensors provide a good balance between cost, availability and accuracy, however, they are generally immobile setups. Recent successes in computer vision and deep learning have made it possible to extract pose data from monocular RGB cameras. This includes approaches that infer 2D pose of one or multiple humans from a single image [11, 12, 46], multiple cameras [14], or estimating 3D poses from 2D images [34].

There also exists specialized external hardware for pose tracking in VR and AR [33]. For example, the HTC Vive [19, 20], Oculus Rift [36] and PlayStation VR [54] track the head, hand controllers and other limb-borne accessories using external sensor base stations. Any un-sensed joints can be roughly estimated with inverse kinematics [47, 50]. Other non-worn, external approaches for pose estimation include capacitive sensing [68], magnetic fields [44, 49], RF [69], and mechanical linkages [55].

2.2 Body Capture with non-IMU Worn Sensors

Pose estimation using body-worn sensors is more portable and flexible than systems requiring external components. Much research has focused on capturing specific body parts. For instance, hand pose is of great importance in VR/AR applications, and has been tracked using e.g., wrist-worn cameras [10, 29, 62], electrical impedance tomography [67], electromyography [30], depth sensors [16], magnetic trackers [13], and specialized gloves [17]. Faces are important too, most often captured using cameras [4, 57], but other methods such as ultrasound [23] and electromyography [18] have also been explored.

Our research is more concerned with *whole-body* pose estimation. Many body-worn sensing approaches exist, ranging from exoskeletons [35], ultrasonic beacons [59], pressure sensors [66] and RFID [27]. Body-worn camera approaches are particularly popular, such as work by Shiratori et al. [52], Ng et al. [42], and Ahuja et al. [2, 3]. All of the latter approaches require specialized additional hardware that most users do not own. The goal of our work is to bring the flexibility of body-worn pose estimation to users without requiring them to purchase any new devices. One prior work with a similar mantra is Pose-On-The-Go [6], which estimates a user's full-body pose using only the sensors in a smartphone and when held in the hand. However, much of the full-body pose is guessed (rather than tracked) by measuring absolute movement and using an animated, rigged (IK) avatar.

2.3 Body Pose Estimation using Worn IMUs

In this section, we focus on pose systems relying exclusively on worn IMUs, which most closely matches our technical approach (Table 1 provides an overview of prior systems). While single IMUs have been used to track individual limbs (such as arm pose in ArmTrack [51]), it is more common to see "fleets" of IMUs distributed across the body (e.g., the popular XSens [63] suit) for full-body pose estimation. Importantly, these setups are homogeneous in terms of IMUs (and thus performance and noise) and tend to use high-quality sensors running at high framerates not typically seen in consumer mobile devices. As we found, the IMUs utilized in Apple's own ecosystem vary by device, and as such, the data varies in quality, noise and framerate.

Among prior work using many body-worn IMUs, we see Tautges et al. [56] able to generate visually plausible motion streams using four XSens IMUs. Sparse Inertial Poser [60] and Deep Inertial Poser [21] use optimization and deep learning-based methods for full-body pose estimation using between 6 and 17 body-worn IMUs. Both systems use SMPL [31, 48], a statistical body model, as their pose output. Approaches such as TransPose [65] or Physical Inertial Poser [64] build on such efforts and provide more accurate representations and better models. All these works leverage the fact that the employed IMUs have known and calibrated positions, and the same noise profiles.

3 POSSIBLE DEVICE COMBINATIONS

Smartphones, smartwatches, and earbuds have different possible body locations. For instance, a smartphone can be stored in the left or right pocket, held in the left or right hand, held to the head (to take a call), or not carried by the user at all (6 possible states). For smartwatches, they are either worn on the left or right wrist or not worn by the user (3 possible states). For earbud-like devices, they can be worn on the head, placed into a charging case and stored in the left or right pocket, or not carried by the user (4 possible states). Although at present, putting earbuds into a charging case generally puts them to sleep, we assume that in the future a firmware update

could allow for continuous IMU data streaming, especially given the larger battery in the case.

Fully enumerated, this yields 72 possible device-location combinations. However, we eliminate three combinations where the user is both wearing earbuds and the phone is held to the head (to take a call), as this is not a typical use case. An additional invalid combination is no phone, smartwatch or earbuds, and thus our system would not run at all. This leaves 68 possible arrangements combinations — 14 combinations have 1 active device, 36 combinations have 2 active devices, and 18 combinations have 3 active devices.

Next, it is important to consider that some combinations of devices do not provide substantially different body data for our purposes. For instance, a user could wear a smartwatch on their left wrist and hold a smartphone in their left hand – the IMU data would be highly correlated, and thus we treat it as a single body point. Another example is storing a smartphone in the right pocket, along with earbuds in a charging case – again, the IMU data would be similar. Thus, what our system truly cares about are the combinations of body location enabled by the 68 possible device-position combinations – these 24 combinations are illustrated in Figure 3.

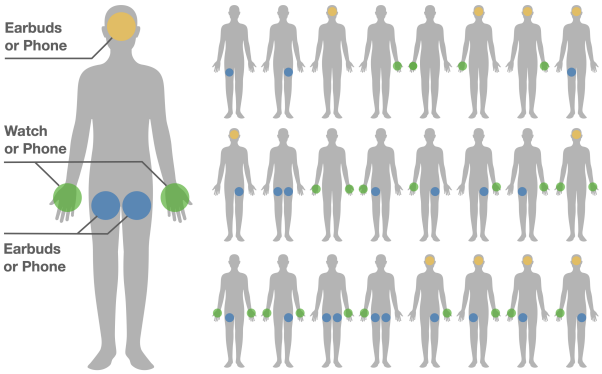

Figure 3: The 24 possible device-location combinations we support and investigate.

We note that our system makes some simplifying assumptions about body positions. For example, in order for a hand position to be considered active, it requires either a smartphone to be held in that hand or a smartwatch worn on that wrist. Even though the signal is not identical, it is highly correlated such that the information power is similar. Similarly, a smartphone held to either ear is considered to be a head location (rather than left or right ear). We made the latter simplification because Apple's AirPods (which we use in our real-time implementation) fuse their IMUs to provide a single-head 6DOF estimate, rather than provide IMU data from each Airpod individually.

4 IMPLEMENTATION

Figure 4 provides an overview of our pipeline. We focus on three popular consumer devices: smartphone, smartwatch and wireless earbuds/headphones. Each of these devices contains an IMU, the ability to wirelessly transmit data, and some local compute. We envision our model executing on the most capable device carried by the user, with the other less-capable devices streaming their IMU data over e.g., Bluetooth.

4.1 Model

For the learning architecture, we use a two-layer Bidirectional LSTM, inspired by prior works [21, 65]. Although we did experiment with newer architectures such as transformers, we found these models did not perform well in practice. LSTMs produced smoother output predictions than the other models we tested. For each available IMU, our system uses orientation (represented as a 3×3 rotation matrix) as well as acceleration as input, both in a global coordinate frame of reference. In contrast to prior work, we do not normalize these inputs to be relative to a root IMU sensor location, such as the pelvis, as our available devices vary. We flatten and concatenate these inputs to form an input vector of size 60: 5 possible IMU locations × (3 acceleration axes + 3×3 orientation rotation matrix), which we input to the model. Of note, our model can ingest any subset of the available IMU data, with absent devices masked (i.e., values set to zero).

The input vector is first transformed into an embedding of dimension 256 using a ReLU [40] activated linear layer. Next, these embeddings are fed sequentially into a Bidirectional LSTM of hidden dimension 256. A final linear layer outputs 144 SMPL [31] pose parameters — 24 joints represented as 6D rotations (which is a smoother representation space [70]). SMPL also provides a body mesh (6890 vertices), which can be seen in Figures 1, 4 and 7. In total, our neural network model has 10.7M trainable parameters. During training, a forward kinematics module calculates joint positions from these pose parameters and further minimizes it with respect to the ground truth.

4.2 IMU Dataset Synthesis

To train our pose model, we required a significant volume of data. For this, we can leverage existing motion capture databases to generate a large synthetic corpus. Specifically, we use AMASS [32], a compilation of 24 motion capture datasets (ACCAD, BioMotion, CMU MoCap, MIXAMO, Human Eva, Human 3.6M, etc.) totaling almost 63 hours of high-quality, high-resolution motion capture data in the SMPL [31] format. A wide variety of motions and activity contexts are included, such as locomotion, sports, dancing,

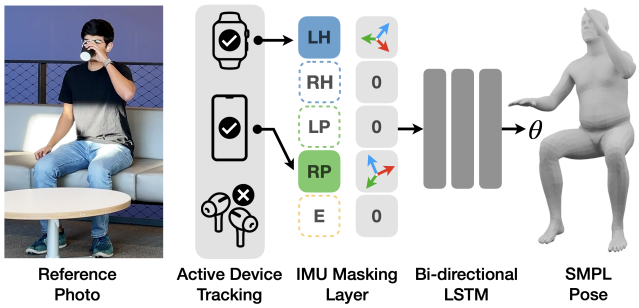

Figure 4: Overview of our real-time system architecture.

exercising, cooking, and freestyle interactions. For additional details on the composition of the AMASS dataset, please refer to [32]. We note that AMASS has been used in much prior work [21, 64, 65] as the basis for deriving synthetic datasets.

The consumer devices we use for our study and real-time demo (described in Sections 5.2.1 and 8) run at a common framerate of 25 FPS. Thus we resample AMASS' 60~120 FPS data to 25 FPS. We then follow the synthetic data generation process used in TransPose [65] and DIP [21]. In short, we "attach" virtual IMUs to specific vertices in the SMPL mesh (the left and right wrists, the left and right front pant pockets, and the scalp) and compute synthetic acceleration data using adjacent frames in the global frame of reference. To generate synthetic orientation data, we calculate joint rotations relative to the global frame by compounding local rotations starting from the joint to the pelvis (root) following the SMPL kinematic chain. We scale acceleration data (m/s^2) by 30 to be suitable for neural networks [65]. Finally, rather than adding synthetic high-frequency noise to our dataset, we instead smooth both synthetic and real-world data using an averaging window of length 5 frames (200 ms), similar to [26].

We use this pipeline to create 24 sets of data, one for each of our 24 device-location combinations (Figure 3), which we combine into a single dataset. We simulate missing devices by masking-out (i.e., zeroing-out) IMU data for those locations. For example, even in our best-case scenario of three devices present, this means that $2/5^{ths}$ of the input vector is null. 63 hours of AMASS data × 25 FPS × 24 device-location combinations yields 134.8M synthetic IMU instances with paired ground truth SMPL poses for training.

4.3 Training

The model is trained end-to-end using PyTorch and PyTorch Lightning deep learning frameworks. We use a batch size of 256 and update the weights using the Adam optimizer with a learning rate of $3e^{-4}$. While training, we use non-overlapping windows of paired IMU and pose data in 5-second (125 samples) chunks. As mentioned earlier, we train our model to regress to full-body pose and full-body joint positions using mean squared error (MSE) loss. Our total loss is the sum of these two individual losses. We train our model for 80 epochs (22 hours) on an NVIDIA Titan X GPU.

4.4 Joint Rotation Refinement

As the last step of our inference pipeline, we adopt the Inverse Kinematic refinement method presented in [25] to perform a final refinement of our output pose. Although our model predicts the rotation of legs, hands and head, it does not necessarily fully honor the absolute orientation offered by the IMUs, even when weighted heavily in our loss term. However, it is logical to take advantage of IMU orientation for limbs with devices, as it is both an absolute value and considerably less noisy than accelerometer data. More specifically, as we have absolute orientation from the IMUs, we optimize certain bone orientations for each instrumented joint. In particular, for the wrist joint (smartwatch/phone), we optimize the elbow and the shoulder orientations, and similarly for the head (earbuds/phone) and hip (smartphone/earbuds case) joints. We implement this using the PyTorch framework and optimize this error using the MSE loss and the Adam optimizer. We allow this optimization to run for 10 iterations on each frame, which we found to not impede real-time performance.

5 EVALUATION

We systematically isolate and analyze the efficacy of IMUPoser across different datasets and conditions.

5.1 DIP-IMU Dataset

To test the performance of our model on real (and not synthetic) IMU data, we use DIP-IMU [21], an IMU-based MoCap dataset. While smaller than the AMASS dataset we used for training, it offers a good variety of poses and activities across five classes: upper-body (arm raises, stretches, swings, etc.), lower-body (leg raises, squats, lunges, etc.), interaction (gestures to interact with everyday objects), freestyle (jumping jacks, punching, kicking, etc.) and locomotion (walking, side steps, etc.). A secondary benefit of using DIP-IMU is that it has been used for evaluation in other similar works [21, 26, 64, 65], permitting direct comparison. DIP-IMU used the commercially-available Xsens [63] IMU-based system to capture data from 10 participants. The data is sampled at 60 Hz, leading to a total dataset size of approximately 90 mins.

5.2 IMUPoser Dataset

As noted above, DIP-IMU used the professional-grade XSens system for data collection, which costs approximately $4000 USD. All of the

Figure 5: IMUPoser data collection setup. Participants wore 2 smartwatches, kept 2 smartphones in their front pockets, and wore wireless earbuds. 41 retroreflective motion capture markers were also placed around the body to track ground truth body pose.

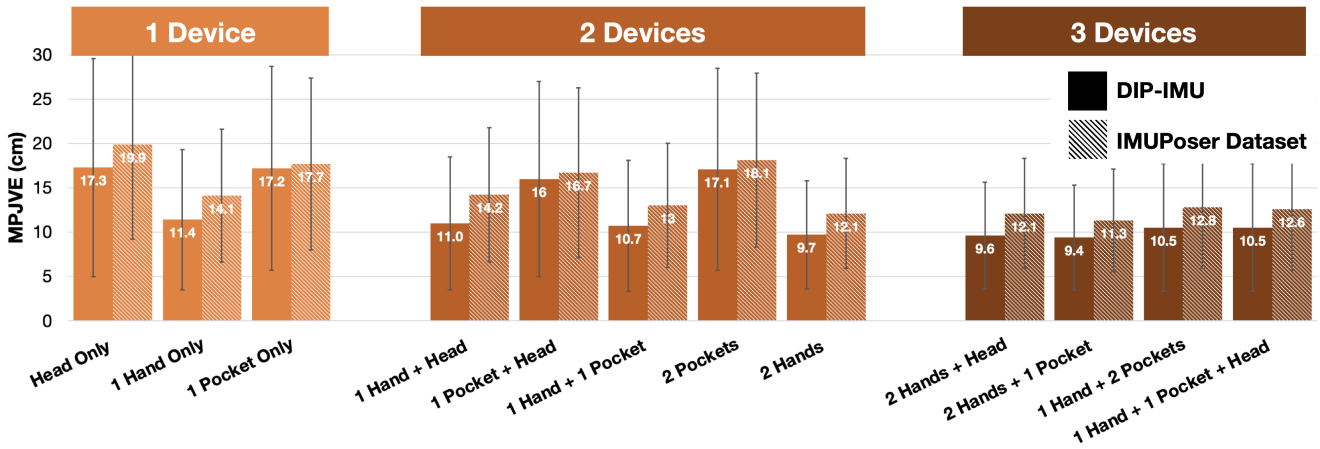

Figure 6: Accuracy across different device combinations. Error is Mean Per Joint Vertex Error (MPJVE) in cm. Note how error decreases as the number of devices increases.

IMUs are matched, offering similar noise and tracking performance. To complement this dataset with a *consumer* device equivalent, we collected our own dataset.

5.2.1 Data Collection Apparatus. Our data collection apparatus consisted of two smartphones (Apple iPhone 11 Pro) placed in the left and right front pockets, two smartwatches (Apple Watch Series 6) placed on the left and right wrists, and one pair of Apple AirPods Pro worn in the ears (Figure 5). The sampling rate of our system was configured to 25 Hz, the maximum sampling rate of the AirPods. The Apple Watch and AirPods communicated over Bluetooth to the iPhones, and the two iPhones relayed all IMU data to a laptop for data processing and recording. Although users had all five devices on them during data capture, we only use a subset of these devices for pose estimation, as described in Section 3.

For ground truth pose, we use a Vicon Motion Capture System system [58] with twelve MX40 cameras and four T160 cameras capturing at 120FPS. We used Vicon Blade 3.2 for capture and data export and Vicon IQ 2.5 for data cleaning. We downsample the Vicon data and synchronize it with our collected IMU data streams. For analysis, we fit an SMPL mesh to the Vicon data using Mosh++ [32].

5.2.2 Device Calibration. In contrast to commercial IMU-based motion capture systems like XSens, smartphones, smartwatches, and earbuds fail to provide IMU orientations in a common (global) frame of reference. If a device contains a magnetometer, the manufacturer usually provides a way to access the orientation of the device in a global frame of reference oriented with Earth's gravitation and magnetic fields. While the iPhones and Apple Watches that we used for this study contained magnetometers, we found their global orientation data to be fairly noisy. Moreover, the Apple AirPods do not contain magnetometers and hence only provide orientation relative to the initial frame of reference of the head. As a result, we opted to use the XArbirtraryCorrectedZVertical frame of reference provided by the Swift CoreMotion API [8].

Before the study began, we aligned all the devices to a common frame of reference and recorded their orientation values over a window of three seconds. This acted as calibration data, bringing all the devices into the same global frame of reference. In practice, since the AirPods only sampled IMU data when they were in a participant's ears, the common frame of reference was set to that. In line with prior works [21, 65], we asked participants to make a T-pose for three seconds to calculate the orientation offsets between the device and the bone joint that it was attached to. The T-pose acts as a template pose wherein rotations are identity and thus known for each joint. This helps calibrate for users wearing the devices in different orientations, for example, a phone held in the hand vs. a watch worn on the wrist.

5.2.3 Data Collection Procedure. For our data collection, we recruited 10 participants (5 identified as female, 5 identified as male) with a mean age of 22. The study lasted roughly 45 minutes and paid $20 in compensation. We asked participants to wear and store the five devices in the way that felt most natural to them. Other than requesting the participants to wear pants with pockets, we did not control for differences in clothing, pocket styles, or smartwatch placement preference on the wrist, so as to get realistic real-world variation. For our Vicon-derived ground truth, we placed 41 optical markers on participants. In order to keep the markers secure, we asked participants to tuck in their shirts and provided velcro straps where needed.

Inspired by prior works [6, 21], we collected our data using an "obstacle course"-style procedure. We extended the classes in the DIP-IMU dataset and included the following motions:

- *Upper Body*: Right arm raises, left arm raises, both arm raises, right arm swings, left arm swings, both arms swinging, arms crossing across the torso, and arms crossing behind the head.
- *Lower Body*: Right leg raises, left leg raises, squats, lunges with left leg, and lunges with right leg.
- *Locomotion*: Walking in a straight line, walking in a figure 8, walking in a circle, sidesteps with legs crossed, and sidesteps with feet touching.
- *Freestyle*: Jumping jacks, tennis swings, boxing with alternate arms, kicking with the dominant leg, push-ups, and dribbling a basketball.

- *Head Motions*: Moving head up-and-down, moving head left-to-right, leaning head from shoulder-to-shoulder, and moving head in circles.
- *Interaction*: Scrolling on a smartphone while seated in a chair.
- *Miscellaneous*: Waving with right arm, waving with left arm, clapping, hopping on right leg, hopping on left leg, jogging in a straight line, and jogging in a circle.

The upper body, lower body, locomotion, freestyle, head motions, interaction and miscellaneous scenarios lasted for 69.7, 43.4, 95.3, 76.2, 36.8, 19.2, and 74.3 seconds on average, respectively, resulting in roughly 7 minutes of data per participant. All the motions were continuous and data was also captured while participants were transitioning from one category to another.

5.3 Evaluation Protocol

In order to compare with prior works, we follow the exact method detailed in [21, 26, 64, 65]. Specifically, we use data from the first eight participants of DIP-IMU as training data, with the last two participants used for testing. We fine-tune our AMASS-trained model using this training data downsampled to 25 Hz to match both our AMASS training data and our real-time system's capabilities. We further test this model on our IMUPoser Dataset, helping to assess real-world accuracy and performance. Our model is evaluated in an online fashion. In particular, we feed a rolling window of 125 samples (5-second history) with a 1-sample overlap, emulating real-world use. This data is smoothed using an averaging filter, as described in Section 4.2. We analyze these results using different evaluation metrics across various device-location combinations. Also following prior work [21, 64, 65], we make use of the following evaluation metrics to quantify the performance of our full-body pose estimation pipeline:

(1) *Mean Per Joint Rotation Error:* MPJRE measures the mean global angular error across all joints in degrees (°).
(2) *Mean Per Joint Position Error:* MPJPE measures the mean Euclidean distance error of all estimated joints in centimeters (cm) with the root joint (pelvis) aligned.
(3) *Mean Per Joint Vertex Error:* MPJVE measures the mean Euclidean distance error across all vertices of the estimated SMPL mesh in centimeters (cm) with the root joint (pelvis) aligned.
(4) *Mean Per Joint Jitter (Jitter):* Jitter measures the average jerk of the predicted motion [65]. A lower jerk value signifies a smoother and more natural motion.

We use mesh error (MPJVE) as our primary evaluation metric for most tasks, due to its ease of understanding and its utility as a benchmark for comparison with prior work.

6 RESULTS

We first describe IMUPoser's accuracy across device-location combinations, before changing our focus to look at results by body region. We conclude this section with a comparison to other related systems.

6.1 Accuracy Across Device-Location Combinations

To simplify presentation of results, we group the 24 possible device-location combinations (Figure 3) into 12 supersets based on the number of devices present and their body locations (ignoring left/right

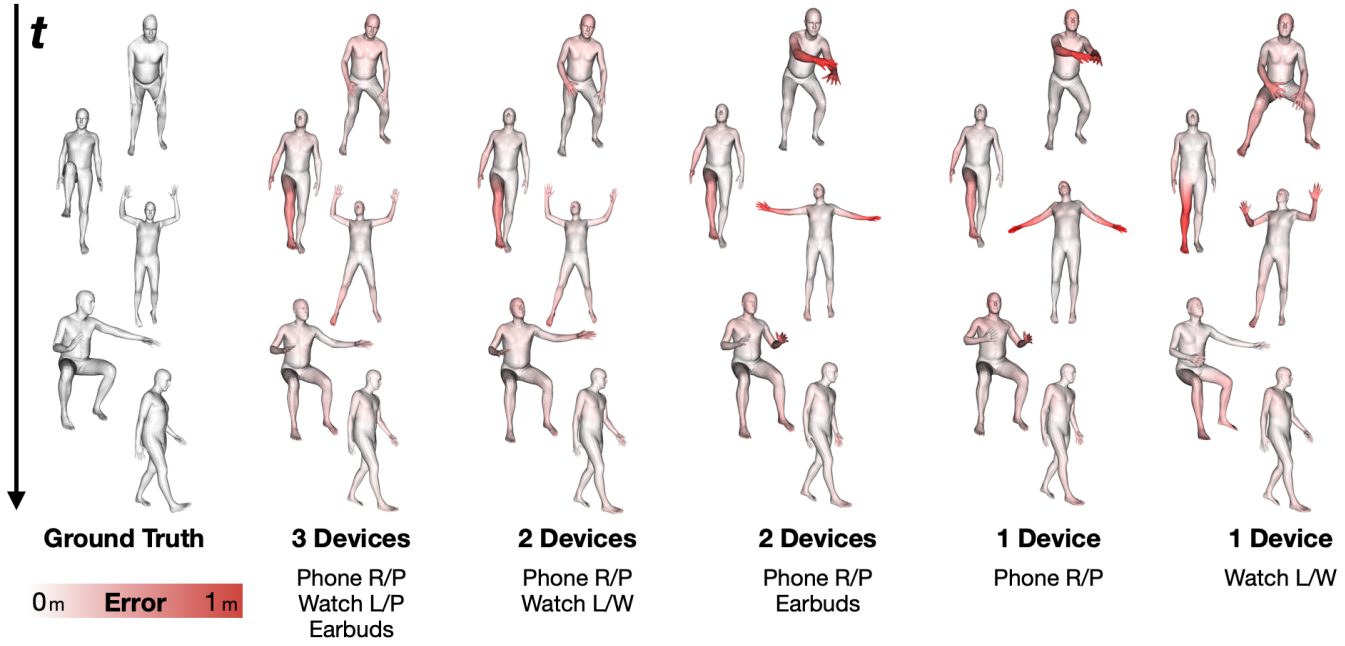

Figure 7: Sample SMPL mesh predictions for different device placements and combinations. The red color indicates the per vertex error in meters (ranging from 0 to 1 m).

placements). Figure 6 presents the results for the IMUPoser and DIP-IMU datasets. Note, that our model has not been fine-tuned on the IMUPoser Dataset. Across all device combinations, we find a MPJVE of 14.1 cm on the IMUPoser Dataset and a MPJVE of 12.1 cm with DIP-IMU. When averaging the results across both datasets, having one device on the user results in a MPJVE of 16.27 cm (SD=9.93 cm), which decreases to 13.9 cm (SD=8.36 cm) when a second device is present. The lowest error, unsurprisingly, is when three devices are present – a MPJVE of 11.1 cm (SD=6.51 cm) across all possible three-device combinations.

Figure 7 offers example mesh predictions across different device-location combinations. As expected, accurate head orientation estimation is only plausible when earbuds are present. Other times the head regresses to the most natural orientation given where the body is facing. Global body orientation works best when at least two devices are present. Lastly, motions that have a characteristic cadence, such as walking, work well across all combinations. Similarly, activities with symmetric limb motions, such as jumping jacks, work fairly well even with no sensor data from important limbs. On the other hand, activities with uncorrelated limb motions fail unless limbs are instrumented.

6.2 Accuracy Across Body Regions

Figure 8 provides a breakdown of system accuracy across different body regions for the IMUPoser and DIP-IMU datasets. We note that the accuracy for a limb with an instrumented point is always greater than that of an uninstrumented one. For example, averaging across both datasets and with an IMU present on the right hand, the MPJVE is 14.65 cm for the right arm (right hand = 17.2 cm) vs. 21.6 cm for the left arm (left hand = 26.9 cm). Unsurprisingly, the highest error is when none of the limbs in a particular body region have IMU data.

Also unsurprising is that the lowest error is achieved when both left and right limbs have IMUs present. For example, with only one IMU on the arms, the MPJVE for both arms is 18 cm (right hand = 22.17 cm; left hand = 20.4 cm). Whereas with both arms having IMUs, the MPJVE is 14.5 cm (right hand = 17.35 cm; left hand = 16.9 cm). A partial exception to this trend is the legs. Unlike the arms, which can move independently, legs tend to move in tandem (out of phase when walking, or in phase for activities such as jumping). This means that even one IMU on the legs is still highly effective at predicting both legs, and two IMUs offer just a modest gain. Looking at our results, the MPJVE for the left leg is 10.3 cm (left foot = 15.65 cm) when the IMU is in the left pocket, and the error for the right leg is 10.4 cm (right foot = 16 cm) when the IMU is in the right pocket. When both IMUs are present (i.e., left and right pockets), the error of the left and right legs drop very modestly to 10.05 cm and 9.75 cm, respectively.

We note that error accumulates along the kinematic chain (see Figure 7). Across all conditions, the average error of the end-effectors (left hand, right hand, left foot, right foot, head) is 20.28 cm and 17.29 cm on the IMUPoser Dataset and DIP-IMU Dataset, respectively (vs. 12.92 cm and 11.23 cm for joints that are not end-effectors).

6.3 Comparison to Prior Work

To the best of our knowledge, no prior research has investigated deriving full-body pose from such a sparse set of consumer-grade devices equipped with IMUs. Table 2 offers a quantitative comparison against key prior work, all evaluated on the same DIP-IMU Dataset [21].

Unsurprisingly, for a system that uses between 1 and 3 IMUs, our model is less accurate than those utilizing 6 sensors (i.e., IMUs placed on each limb). However, compared to DIP [21] and Transpose [65], our MPJVE is only worse by 3.2 cm and 5.0 cm, respectively. It is interesting to note that the Jitter of our system is in line with prior work (1.9 vs. 1.4 of TransPose). At a high level, even with an impoverished sensing configuration, we are able to produce natural, realistic and smooth pose estimation sequences. In the future, we hope to combine physics-backed models (as in PIP) to further improve the pose estimation of our system.

7 ACTIVE DEVICE TRACKING

A crucial piece of information our pose model needs before it can run is: 1) What devices are present on the user? And 2) where these devices are located on the body? For this, we created a separate piece of software, which runs in parallel with our pose model.

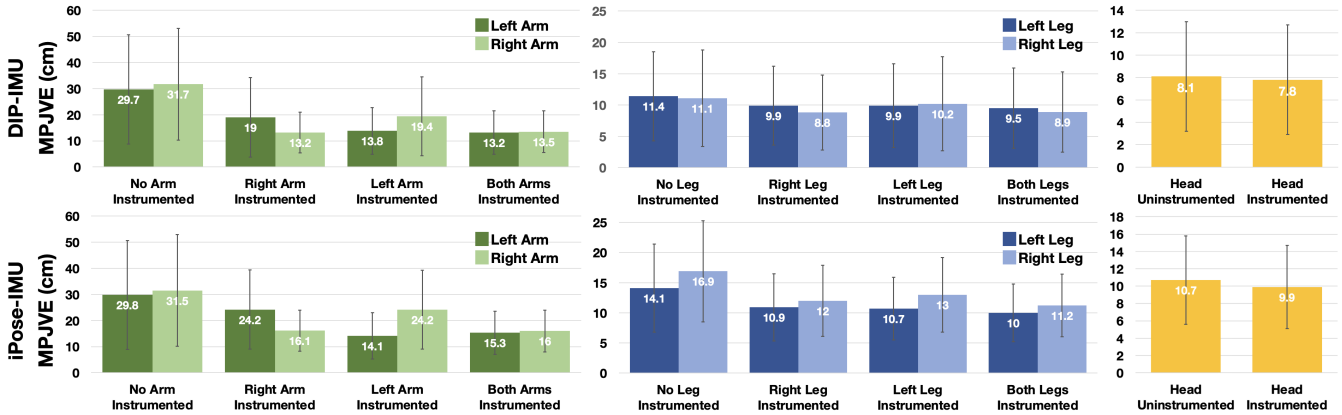

Figure 8: Summarized accuracy results across different body regions evaluated on the DIP-IMU and IMUPoser datasets.

System	# Inst. Joints	MPJRE (°)	MPJPE (cm)	MPJVE (cm)	Jitter ($10^2 m/s^3$)
SIP (offline)	6	8.7	6.7	7.7	3.8
DIP (online)	6	15.1	7.3	8.9	30.13
TransPose (online)	6	8.8	5.9	7.1	1.4
PIP (online)	6	-	-	5.9	0.24
IMUPoser (online)	1–3	23.9	9.7	12.1	1.9

Table 2: Comparison of IMUPoser to key prior work, all evaluated on the DIP-IMU Dataset [21].

7.1 Implementation

To determine where devices are located on the body, we require three pieces of information from the user, which we envision being collected when a user first purchases a device. 1) In which pocket do they typically store their phone? 2) In which hand do they typically hold their phone? And 3) On which arm would they wear a smartwatch? After this basic initialization, we use a series of automated heuristics.

We make the assumption a smartphone is held in the hand if the screen is on and the IMU is reporting even slight motion. If the user is wearing a smartwatch, we can use the distance between the watch and phone (provided by Apple's NINearbyObject API [9], which uses UWB) to guess the holding hand automatically (see Figure 10). If no smartwatch is worn, our system falls back on the hand specified by the user during setup. If the smartphone screen is off and the IR proximity sensor is triggered, we assume the phone is in a pocket. If the user has a smartwatch, we can similarly use UWB-derived distance to guess the pocket. If no smartwatch is worn, we default to the user-specified pocket.

As most users wear watches in a consistent location, the logic for smartwatches is simpler. If it is connected to the iPhone and moving, we assume it is worn on the user-specified hand. Similarly, for Airpods, if they are connected to the iPhone, we know they are in the ear. When in their charging case, Airpods go to sleep and stop transmitting IMU data. However, we believe that Apple could modify the Airpods firmware such that in the future they could continue to transmit IMU data even when stored in a pocket.

7.2 Evaluation

As a preliminary evaluation of our active device tracking prediction, we ran a user study with 7 participants (5 identified as male, 2 identified as female; mean age 27.8; all right-handed with a preference for wearing watches on the left wrist). The study lasted approximately 15 minutes and paid $5. To initialize our system, we recorded participants' answers to the three preference questions listed in the previous section. We then asked users to transition between 15 device combinations, in a random order, documented in Figure 9. When a device was not in a requested set, it was set aside on a nearby table. For each requested device-location combination, we asked participants to walk around for about 10 seconds, then sit down briefly, rise to stand again, and lastly return to the starting position. Before the next trial began, the necessary devices were given or taken from the participants. Throughout the study, our active device tracking process ran, making live predictions about what devices were active and where they were located. A trained experimenter conducted the study, marking the start and stop of each device combination trial, alongside ground truth labels.

Across all participants and all data instances, the accuracy of earbuds and smartwatch tracking was 100%, owing to their known locations and very reliable detection of worn vs. not worn. Smartphone tracking is the most challenging, with five possible states (not present, left pocket, right pocket, left hand, right hand). We found the instance-wise accuracy for smartphone tracking was 90.8%.

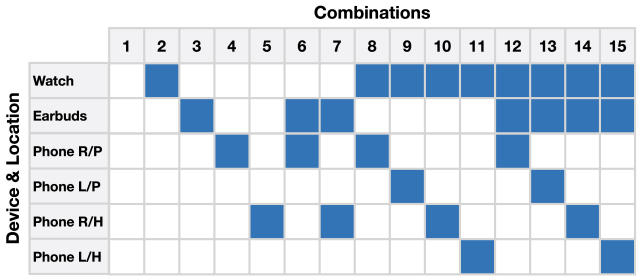

Figure 9: Device combinations tested as part of our active device tracking study. Blue denotes presence in the set.

8 REAL-TIME IMPLEMENTATION

To help demonstrate the imminent feasibility of our approach, we created a real-time implementation of IMUPoser, which can be seen in our Video Figure. It is comprised of two main processes working together. First is active device tracking, which monitors what devices are available to provide IMU data and predicts where they are located on the body. Second is our pose model, which is passed the location inferences and IMU data.

8.1 Proof-of-Concept Device Ecosystem

As a proof-of-concept implementation, we use an Apple iPhone 11 Pro, Apple Watch Series 6, and AirPods Pro. Apple offers a mature inter-device API that allows these devices to exchange data. Each device reports 6DOF IMU data at different rates, with the slowest being AirPods at roughly 25 FPS. We also note that although AirPods come as a pair, they fuse their individual IMUs into a single 6DOF head estimate.

8.2 Output

As a proof of concept, we use an iPhone optionally connected to an Apple Watch and AirPods. The iPhone streams all available IMU

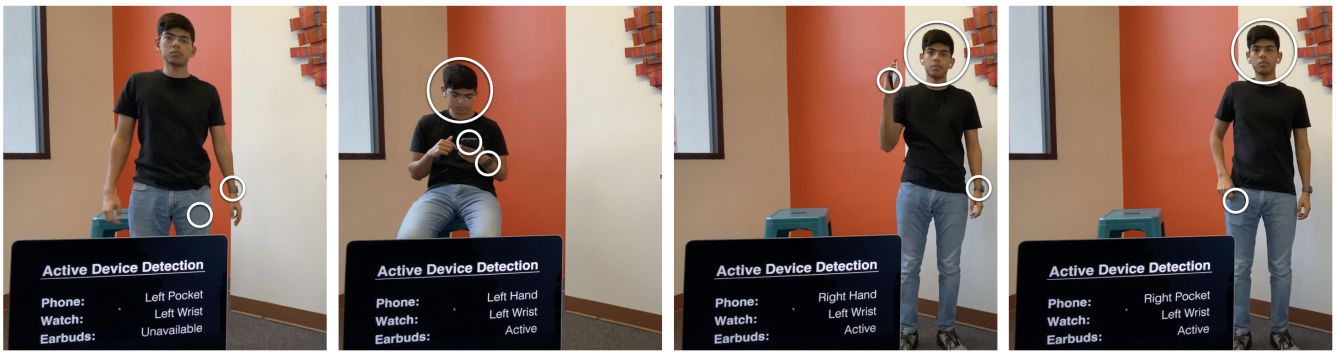

Figure 10: Active device tracking across different device combinations. Active devices are highlighted with a white circle for illustration and the foreground laptop shows the live tracking result.

data back to a MacBook Air (2021), which runs our active device tracking and pose estimation processes, with a mean inference time of 26.8 ms. We believe our model could be run on a mobile phone with additional engineering effort. Regardless of where the model runs, it is capped at 25 FPS, the reporting frequency of our slowest IMU (Airpods). Before running our system, we must perform the same calibration as in data collection (see Section 5.2.2). For real-time output, we visualize the SMPL mesh.

9 OPEN SOURCE

To enable other researchers and practitioners to build upon our system, we have made our dataset, architecture, trained models, and visualization tools freely available at https://github.com/FIGLAB/IMUPoser with the gracious permission of our participants.

10 LIMITATIONS AND FUTURE WORK

While IMUPoser enables pathways to full-body pose estimation with minimal user instrumentation, it has pros and cons like any other technical approach. While IMUPoser can glean insights about the pose of limbs for which it has no direct sensor data, it is important to note that such a pose is only an approximate result. For cases where the motion of the instrumented joint is completely independent of that of the uninstrumented one, IMUPoser tends to regress to the mean pose. IMUPoser can support the incorporation of new joint locations by using the corresponding SMPL mesh vertices for training. Thus, in the future, IMUPoser can potentially support and track new device placements, such as a phone in a back pants pocket, coat pocket, armband, etc. The fidelity of the system could also be improved by integrating additional consumer devices (e.g., smart shoes, eye-wear, rings) into the ecosystem. This would help expand the range of poses supported by IMUPoser, allowing it to track dynamic activities such as cycling, kayaking, skiing, etc.

Unlike Transpose [65] and PIP [64], the current implementation of IMUPoser does not predict global root translation. In the future, using better learning methods and multimodal cues when available (e.g., visual odometry from the smartphone [6]) could help predict translation. Along similar lines, the overall accuracy of the system could be improved by including contextual cues such as the activity being performed [7] or the user's location.

Another limitation of our system is active device tracking. Currently, this is a basic, proof-of-concept implementation that needs further refinement before it can be deployed for consumer use. Furthermore, all of the devices need to be in a homogeneous ecosystem (e.g., Apple) to work effectively. In the future, the use of a common industry-wide standard to connect and network between different consumer devices can help mitigate this issue.

Finally, we envision IMUPoser executing on the most capable device the user happens to be carrying. In most cases, this will be a smartphone, but not always, especially in the future. For instance, it is possible today for a user to go for a run with a smartwatch and wireless headphones, but no phone. In the near future, it seems possible there will be AirPod-like devices that can operate independently (e.g., hearables).

11 CONCLUSION

In this paper, we presented IMUPoser – a system for real-time, full-body pose estimation using IMUs present in consumer devices such as phones, smartwatches and earbuds. Our system must automatically track devices that are available and where they are currently located on the body, and use streaming IMU data to estimate pose. Our evaluations show that IMUPoser can contend with the noisy signals of consumer IMUs and produce natural and temporally-coherent pose estimates with as little as one device. This opens up new and interesting whole-body applications with no additional user instrumentation.

ACKNOWLEDGMENTS

We would like to thank Justin Macey and the Computer Graphics Lab at Carnegie Mellon University for assisting with collecting and processing the IMUPoser Dataset. We are also grateful to Giorgio Becherini and the Perceiving Systems group at the Max Planck Institute for Intelligent Systems for help with processing our motion capture data.

REFERENCES

[1] Vida Adeli, Ehsan Adeli, Ian Reid, Juan Carlos Niebles, and Hamid Rezatofighi. 2020. Socially and contextually aware human motion and pose forecasting. *IEEE Robotics and Automation Letters* 5, 4 (2020), 6033–6040.

[2] Karan Ahuja, Mayank Goel, and Chris Harrison. 2020. BodySLAM: Opportunistic User Digitization in Multi-User AR/VR Experiences. In *Symposium on Spatial User Interaction*. 1–8.

[3] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/3332165.3347889

[4] Karan Ahuja, Rahul Islam, Varun Parashar, Kuntal Dey, Chris Harrison, and Mayank Goel. 2018. Eyespyvr: Interactive eye sensing using off-the-shelf, smartphone-based vr headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–10.

[5] Karan Ahuja, Andy Kong, Mayank Goel, and Chris Harrison. 2020. Direction-of-Voice (DoV) Estimation for Intuitive Speech Interaction with Smart Devices Ecosystems *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1121–1131. https://doi.org/10.1145/3379337.3415588

[6] Karan Ahuja, Sven Mayer, Mayank Goel, and Chris Harrison. 2021. Pose-on-the-Go: Approximating User Pose with Smartphone Sensor Fusion and Inverse Kinematics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[7] Karan Ahuja, Eric Whitmire, Joseph Greer, and Wolf Kienzle. 2022. ActivityPoser: Activity driven Full-Body Pose Estimation from Sparse IMU Configurations. In *Symposium on Spatial User Interaction*. 1–2.

[8] Apple. 2022. CMAttitudeRef. https://developer.apple.com/documentation/coremotion/cmattitudereferenceframe

[9] Apple. 2022. NINearbyObject. https://developer.apple.com/documentation/nearbyinteraction/ninearbyobject

[10] Riku Arakawa, Azumi Maekawa, Zendai Kashino, and Masahiko Inami. 2020. Hand with Sensing Sphere: Body-Centered Spatial Interactions with a Hand-Worn Spherical Camera. In *SUI '20: Symposium on Spatial User Interaction, Virtual Event, Canada, October 31 - November 1, 2020*. ACM, New York, 1:1–1:10. https://doi.org/10.1145/3385959.3418450

[11] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[12] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR '17)*. IEEE, 7291–7299. https://doi.org/10.1109/CVPR.2017.143

[13] Ke-Yu Chen, Shwetak N Patel, and Sean Keller. 2016. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1504–1514.

[14] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. 2008. Performance capture from sparse multi-view video. *ACM transactions on graphics* 27, 3 (Aug. 2008), 1–10. https://doi.org/10.1145/1360612.1360697

[15] Yann Desmarais, Denis Mottet, Pierre Slangen, and Philippe Montesinos. 2021. A review of 3D human pose estimation algorithms for markerless motion capture. *Computer vision and image understanding: CVIU* 212 (Nov. 2021), 103275. https://doi.org/10.1016/j.cviu.2021.103275

[16] Nathan Devrio and Chris Harrison. 2022. DiscoBand: Multiview Depth-Sensing Smartwatch Strap for Hand, Body and Environment Tracking. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 56, 13 pages. https://doi.org/10.1145/3526113.3545634

[17] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

[18] Anna Gruebler and Kenji Suzuki. 2014. Design of a Wearable Device for Reading Positive Expressions from Facial EMG Signals. *IEEE Transactions on Affective Computing* 5 (2014), 227–237.

[19] HTC. 2020. VIVE. https://www.vive.com

[20] HTC. 2020. VIVE Accessory Trackers. https://www.vive.com/us/accessory/vive-tracker

[21] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

[22] Intel Corporation. 2020. RealSense. https://www.intelrealsense.com/

[23] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. 2019. Interferi: Gesture sensing using on-body acoustic interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[24] Ahmad Jalal, Md Zia Uddin, and T-S Kim. 2012. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics* 58, 3 (2012), 863–871.

[25] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. 2022. AvatarPoser: Articulated Full-Body Pose Tracking from Sparse Motion Sensing. *arXiv preprint arXiv:2207.13784* (2022).

[26] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W. Winkler, and C. Karen Liu. 2022. Transformer Inertial Poser: Attention-based Real-time Human Motion Reconstruction from Sparse IMUs. https://doi.org/10.48550/ARXIV.2203.15720

[27] Haojian Jin, Jingxian Wang, Zhijian Yang, Swarun Kumar, and Jason Hong. 2018. Rf-wear: Towards wearable everyday skeleton tracking using passive rfids. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 369–372.

[28] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam: Detecting, recognizing and tracking simultaneous exercises in unconstrained scenes. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–17.

[29] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 167–176.

[30] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D Hand Pose Tracking Using EMG Wearables. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1471–1482. https://doi.org/10.1145/3442381.3449890

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: a skinned multi-person linear model. *ACM transactions on graphics* 34, 6 (Oct. 2015), 1–16. https://doi.org/10.1145/2816795.2818013

[32] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5442–5451.

[33] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. 2016. Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE transactions on visualization and computer graphics* 22, 12 (Dec. 2016), 2633–2651. https://doi.org/10.1109/TVCG.2015.2513408

[34] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

[35] Meta Motion. 2018. Gypsy Motion Capture System. http://metamotion.com/gypsy/gypsy-motion-capture-system.htm

[36] Meta Technologies LLC. 2020. Oculus Rift. https://www.oculus.com/rift-s/

[37] Damien Michel, Ammar Qammaz, and Antonis A Argyros. 2017. Markerless 3d human pose estimation and tracking based on rgbd cameras: an experimental evaluation. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. 115–122.

[38] Microsoft Corporation. 2010. Microsoft Kinect Games. Retrieved 2021 from https://en.wikipedia.org/wiki/Category:Kinect_games

[39] Hossein Mousavi Hondori and Maryam Khademi. 2014. A review on technical and clinical impact of microsoft kinect on physical therapy and rehabilitation. *Journal of medical engineering* 2014 (2014).

[40] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

[41] NaturalPoint Inc. 2020. OptiTrack. http://optitrack.com

[42] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. 2020. You2me: Inferring body pose in egocentric video via first and second person interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9890–9900.

[43] Thong Duy Nguyen and Milan Kresovic. 2022. A survey of top-down approaches for human pose estimation. (Feb. 2022). arXiv:2202.02656 [cs.CV] http://arxiv.org/abs/2202.02656

[44] Northern Digital Inc. 2020. trakSTAR. https://www.ndigital.com/msci/products/drivebay-trakstar

[45] OpenNI. 2020. OpenNI. https://structure.io/openni

[46] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. 2018. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV '18)*. 269–286. https://doi.org/10.1007/978-3-030-01264-9_17

[47] Mathias Parger, Joerg H. Mueller, Dieter Schmalstieg, and Markus Steinberger. 2018. Human Upper-Body Inverse Kinematics for Increased Embodiment in Consumer-Grade Virtual Reality *(VRST '18)*. Association for Computing Machinery, New York, NY, USA, Article 23, 10 pages. https://doi.org/10.1145/3281505.3281529

[48] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 10975–10985.

[49] Polhemus. 2020. Polhemus Motion Capture System. https://polhemus.com/case-study/detail/polhemus-motion-capture-system-is-used-to-measure-real-time-motion-analysis

[50] Root Motion. 2020. FINAL IK - VRIK Solver Locomotion. http://www.root-motion.com/finalikdox/html/page16.html

[51] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*. 85–96.

[52] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K Hodgins. 2011. Motion capture from body-mounted cameras. In *ACM SIGGRAPH 2011 papers*. 1–10.

[53] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. 2013. Real-time human pose recognition in parts from single depth images. *Commun. ACM* 56, 1 (Jan. 2013), 116–124. https://doi.org/10.1145/2398356.2398381

[54] SONY. 2020. PlayStationVR. https://www.playstation.com/en-us/ps-vr

[55] Ivan E. Sutherland. 1968. A Head-Mounted Three Dimensional Display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I* (San Francisco, California) *(AFIPS '68 (Fall, part I))*. Association for Computing Machinery, New York, NY, USA, 757–764. https://doi.org/10.1145/1476589.1476686

[56] Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. 2011. Motion reconstruction using sparse accelerometer data. *ACM Transactions on Graphics (ToG)* 30, 3 (2011), 1–12.

[57] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2018. FaceVR: Real-time gaze-aware facial reenactment in virtual reality. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–15.

[58] Vicon Motion Systems Ltd. 2020. Vicon. https://vicon.com

[59] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. 2007. Practical motion capture in everyday surroundings. *ACM transactions on graphics (TOG)* 26, 3 (2007), 35–es.

[60] Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. 2017. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 349–360.

[61] Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. 2012. Accurate realtime full-body motion capture using a single depth camera. *ACM transactions on graphics* 31, 6 (Nov. 2012), 1–12. https://doi.org/10.1145/2366145.2366207

[62] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M Kitani. 2020. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1147–1160.

[63] Xsens. 2020. Xsens Motion Capture. https://www.xsens.com/motion-capture

[64] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. 2022. Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors. *arXiv preprint arXiv:2203.08528* (2022).

[65] Xinyu Yi, Yuxiao Zhou, and Feng Xu. 2021. TransPose: real-time 3D human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.

[66] KangKang Yin and Dinesh K Pai. 2003. FootSee: an interactive animation system.. In *Symposium on Computer Animation*. Citeseer, 329–338.

[67] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 167–173.

[68] Yang Zhang, Chouchang Yang, Scott E Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++ room-scale interactive and context-aware sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–15.

[69] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE, 7356–7365. https://doi.org/10.1109/CVPR.2018.00768

[70] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2018. On the Continuity of Rotation Representations in Neural Networks. https://doi.org/10.48550/ARXIV.1812.07035

[71] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 2018. 3d human pose estimation in rgbd images for robotic task learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1986–1992.